

AD-A101 148

NAVAL POSTGRADUATE SCHOOL MONTEREY CA
STAR - COPPERHEAD INTERFACE.(U)
MAR 81 T E CAHILL

F/8 19/1

UNCLASSIFIED

NL

1 of 1
AD A
101148

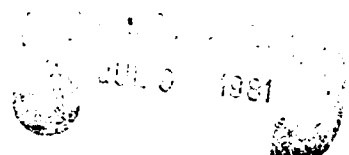
END
DATE
FILMED
7-81
DTIC

11 2
NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD A101493



THESIS



A

STAR - Copperhead Interface

by

Thomas Everett Cahill

March 1981

Thesis Advisor:

A. L. Schoenstadt

Approved for public release; distribution unlimited.

DTIC FILE COPY

01 1 00 012

target selection, firing, and impact phases of the Copperhead system. Interface points of the new logic with the current STAR model are identified and the modifications required to support the new system are discussed. Also presented is an overview of STAR's current Field Artillery module. Key Copperhead system characteristics which influence the model are also described.

Approved for Public Release: Distribution Unlimited

STAR - Copperhead Interface

by

Thomas Everett Cahill
Captain, United States Army
BS, United States Military Academy, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL

March, 1981

Author:

Thomas E. Cahill

Approved by:

Arthur G. B...

Thesis Advisor

J. H. Parry

Second Reader

J. K. Huntman

Chairman, Department of Operations Research

Dean of Information and Policy Sciences

ABSTRACT

This thesis provides the general design logic for a computer representation of the Field Artillery's precision guided munition- Copperhead. The design has been specifically structured to enable its integration into the Simulation of Tactical Alternative Responses (STAR) Model. (STAR is a stochastic force-on-force combat simulation.) Routines and events are developed which portray the target identification, target selection, firing, and impact phases of the Copperhead system. Interface points of the new logic with the current STAR model are identified and the modifications required to support the new system are discussed. Also presented is an overview of STAR's current Field Artillery module. Key Copperhead system characteristics which influence the model are also described.

TABLE OF CONTENTS

I.	INTRODUCTION	9
II.	DESCRIPTION OF STAR FIELD ARTILLERY MODULE	11
III.	THE COPPERHEAD SYSTEM	19
IV.	MODEL DESIGN CONCEPT AND ASSUMPTIONS	25
V.	TERMINAL GUIDANCE AND IMPACT ROUTINES	32
	A. CH.ACQUIRE	34
	B. CH.HIT	36
	C. CH.LASE	39
VI.	INTERFACE EVENTS AND ROUTINES	48
	A. COPPERHEAD	49
	B. CH.FIRE	56
	C. CH.BAL.ERROR	58
VII.	SUMMARY AND FUTURE MODEL ENHANCEMENTS	60
APPENDIX A	CURRENT STAR FIELD ARTILLERY	64
	EVENTS AND ROUTINES	
APPENDIX B	CURRENT STAR FIELD ARTILLERY	71
	INPUT VARIABLES	
APPENDIX C	KEY COPPERHEAD ENTITIES	76
APPENDIX D	COPPERHEAD ARRAYS AND VARIABLES	79
	LIST OF REFERENCES	87
	INITIAL DISTRIBUTION LIST	89

LIST OF TABLES

1. FA MODULE SELECTED ROUTINES/EVENTS	13
2. COPPERHEAD STRENGTHS AND WEAKNESSES	24
3. COPPERHEAD MESSAGE TRAFFIC	54

LIST OF FIGURES

1. COPPERHEAD FOOTPRINT	21
2. STAR-COPPERHEAD INTERFACE	26
3. COPPERHEAD ACQUISITION	33
4. ROUTINE CH.ACQUIRE	35
5. ROUTINE CH.HIT	37
6. EVENT CH.LASE	43
7. ROUTINE COPPERHEAD	50
8. ROUTINE CH.FIRE	57
9. COPPERHEAD BALLISTIC ERROR	59
10. ARRAY AIM.PT DIAGRAM	83
11. ARRAY CH.PKH DIAGRAM	84
12. ARRAY R.HT.FP DIAGRAM	85
13. ARRAY TEMPLATE DIAGRAM	86

ACKNOWLEDGEMENT

I would like to express my appreciation to Professor Art Schoenstadt whose guidance has made this thesis effort a worthwhile learning experience. I am also very grateful to LTC Ed Kelleher for his counsel and assistance in researching the SFAR Field Artillery module. Special thanks are extended to Major Doug Williams and Major Darrell Morgeson for the advice and extra instruction they provided. Finally, I'd like to formally thank my wife, Donna, for her great help.

I. INTRODUCTION

"Of these... weapons systems, Copperhead seems to offer the greatest potential for change in the structure of our fire delivery system. Moreover, when the present tube artillery cannon-launched guided projectile is supplemented by air-dropped, rocket-launched, and mortar fired precision guided munitions, the new weaponry may dramatically alter the balance of combat and restore the advantage to the defense- for the first time since 1939."

[4, p.58]

Traditionally the U.S. Army has always placed heavy reliance on American technology to overcome the numerical superiority of its potential adversaries. This trend has become pronounced in recent years. A crucial dilemma faced today by U.S. military planners is to wisely choose from among many costly, complex, state-of-the-art weapons systems the ones which will maximize marginal return in terms of additional combat power. Moreover, the military planner must determine how doctrine and organization should be modified to best incorporate this new technology into the total force. It is in this arena that the high resolution combat simulation model has become valuable. The purpose of this thesis is, in a general sense, to develop such a model for the field artillery's precision guided munition - Copperhead. The project is intended to ultimately provide planners and tacticians a means to gain insight into this system's capability as a combat force multiplier.

The specific goal of this research effort is composed of two mutually supporting objectives. The first is to develop the simulation logic "modules" which model the Copperhead engagement sequence, especially the final critical phase of target illumination and terminal homing. The second objective is to locate and modify the specific interface points in the Naval Postgraduate School's Simulation Of Tactical Alternative Responses (STAR) model which will enable it to

incorporate these Copperhead modules. Completion of these two objectives provides the logic capable of embedding Copperhead into the indirect fire system of a force-on-force simulation model. This in turn, when implemented, will provide planners with a valuable analytic tool to study the direct and synergistic effects of the Copperhead system. The following paragraph outlines the rest of this report.

Chapter II of this thesis gives a brief summary of the Simulation of Tactical Alternative Responses model, particularly with regard to the current status of its field artillery routines. Chapter III provides a discussion of Copperhead, to include a description of its major components and of the tactical/technical characteristics unique to this projectile. Chapter IV contains a brief overview of the modeling methodology used in this thesis. It describes in general terms the interaction of the current field artillery module with the proposed Copperhead routines. It is designed to provide the reader with a frame of reference from which he can progress to the more detailed discussions in succeeding chapters. Finally, it identifies the major assumptions used in this thesis. Chapter V analyzes the methodology used to model the impact phase of the Copperhead projectile. It should be noted here that most of the modeling concepts used in this phase were derived from a U.S. Army Materiel Systems Analysis Activity technical report [12], written by Mr. Michael Starks, which models the probability of hit as the product of probability of seeker acquisition and the probability of round maneuver. Chapter VI describes interface routines and events which generate Copperhead missions, model the required communication traffic, and simulate the battery firing. Finally, Chapter VII provides a thesis summary and suggests ideas for future model enrichment.

II. DESCRIPTION OF STAR'S FIELD ARTILLERY MODULE

Before an effective discussion of integrating the Copperhead system into STAR can begin some brief background information on the current STAR Field Artillery methodology should be provided. This chapter gives a brief summary.

The Field Artillery module was designed initially as a component part of the Simulation Of Tactical Alternative Responses (STAR) model which has been in existence at the Naval Postgraduate School since December 1978. STAR is a high resolution, stochastic, force on force combat simulation written in Simgscript which plays a blue brigade (or subset) against a red division (or subset). For the reader unfamiliar with the STAR model see references [1], [6], and [11].

The FA module as it currently stands is an evolution of the computer code originally developed by Kelleher [9] and Starner [13], and further expanded by Tradoc Research Element Monterey (TREM) personnel (primarily LTC E. P. Kelleher). It simulates the interacting events which occur when the field artillery system supports the maneuver of a combined arms force. In its present form the FA module plays both blue and red artillery systems up to a level consistent with the corresponding maneuver force size. Direct Support, General Support, and Reinforcing artillery units may be designated by the user and related missions may be simulated. User input also establishes the number and organization of various field artillery (and Simgscript) entities such as forward observer, battery, fire direction center, and fire support officer. The module itself will then maintain these proper functional and organizational relationships during the conduct of subsequent fire missions.

The heart of the FA module is event 'UPDATE.CLUSTERS'

which is the program mechanism that generates targets and initiates the proper sequence of routines/events in the call-for-fire. (See Table 1 for a brief reference guide for selected FA Module routines and events.) Presently there exist 3 mission classifications: a final protective fire, a trigger area, and a standard target of opportunity. Specific final protective fire zones are simulated by ellipses in the battle area and may belong to pre-designated forward observers. When the program detects a user specified number/weighting value of enemy vehicles in a FPF ellipse, it initiates a FPF mission (routine 'URGENT') which has the highest "priority" of the mission classifications. The trigger area mission (also defined as "priority") ranks second in importance as a classification and is also modeled by an ellipse. Its physical location on the battlefield is again predesignated by the user while the activation of the mission (routine 'TRGR') occurs in a manner similar to FPF above. The trigger area was designed in the short term to simulate a field artillery pre-planned target placed on a critical grid, such as a choke point or a likely enemy avenue of approach. In addition, the 'TRGR' logic established a mechanism to later incorporate new technology, such as field artillery scatterable mines, in the model. A target of opportunity is the last mission classification and is processed as routine rather than priority. The computer code associated with the segment generates a fire mission when a predetermined number of enemy vehicles contained on a forward observer's target list become grouped within a relatively small area. (See Appendix A, routine 'DOING.CLUSTERS'). The ranking differences among different classifications of missions are significant since a priority mission is given a shorter system response time than a routine mission and may override (jump ahead in a logic queue) lower ranked missions actually being "fired".

TABLE 1: FA MODULE SELECTED ROUTINES/EVENTS

<u>ROUTINE/EVENT</u>	<u>FUNCTION</u>
BTRY.FIRING (E)	Computes center of impact of an artillery volley and schedules assessment routines.
CZECH.GUNS (E)	Processes missions arriving at battery by sending to guns or placing in howitzer queue.
COMRECD (E)	Key communications routine which initiates specific actions in response to received radio message.
DOING.CLUSTERS (E)	Generates standard target of opportunity mission.
ELL.CHECK (R)	Combines with routine FA.CONV to determine if a target is within an elliptically shaped area.
END.OP. MISSION (E)	Makes final disposition of a fire mission.
FA.ASSESS (R)	Identifies potential victims of artillery fire and calls routine ATRIT to assign specific damage.
FA.CONV (R)	See ELL.CHECK
PSCORD (E)	Prevents duplicate misssions from being processed, calls routine MOA to determine availability of firing units, and assigns batteries to fire specific missions.
INCOMING (R)	Acts as an intermediate routine in the impact assessment phase. Develops the geometrical parameters later sent to routine FA.ASSESS.
MOA (R)	Scans supporting artillery units to determine if user defined 'best method' of attack is possible. If not, user input alternative methods are checked.
NEW.MISSION (E)	Creates the temporary entity MISSION for a standard target of opportunity.
PRI.MSSN (R)	Creates the temporary entity MISSION for a final protective fire or trigger mission.
TRGR (R)	Generates a fire mission when sufficient targets move into a trigger area.

TABLE 1 (continued)

UPDATE.CLUSTERS	(E)	Sequentially calls mission generating routines URGENT, TRGR, and DOING.CLUSTERS.
URGENT	(R)	Generates a fire mission when sufficient targets move into the FO's final protective fire area.

The routines mentioned above, in association with the 'UPDATE.CLUSTER' event, are the most established and verified portion of the FA module. The missions generated by these routines simulate those which would normally be called in by an FO whose primary responsibility is in support of a maneuver element. However, by heavily weighting specific types of enemy vehicles/weapons systems, a user may (if desired) force the FO to initiate counterbattery (CBTRY) or suppression of enemy air defense (SEAD) missions before those of maneuver support. Such a design enables the entire spectrum of mission types to be played in the simulation. Though not yet finished, work is underway which will allow fire missions to be generated by counterbattery radars. For example routine 'RDR.SECTION' identifies all enemy firing batteries which lie within the fan of friendly force radar. Again, the computer code is currently in place within 'FSCORD' and routine 'MOA' to support the CBTRY and SEAD roles. The final method of generating artillery fires is the event 'SCHED.ART.FIRES', which allows the user to initiate indirect fires for either side at a preselected time during the battle.

Damage assessment in the field artillery module is another key segment of the code and is initiated in the event 'BTRY.FIRING'. Within this event an aim point for a selected mission is designated based on its classification. The phase of the mission could be in either 'adjust' or 'fire for effect'. The aim point coordinate is modified by adding a normal delta x and a delta y value to simulate ballistic error. This adjusted aim point is then sent to event 'INCOMING' which uses it as the center of impact of a sheaf or volley of artillery fire. This 'sheaf' is currently modeled as a predetermined elliptical area of lethality but will eventually become a look up table returning sheaf geometry as a function of weapon caliber, number of tubes, and

type. Finally, event 'INCOMING' calls routine 'FA.ASSESS', sending to it the following parameters: the coordinates of the center of the sheaf (aim point), the values for the semi-major and semi-minor axes of the sheaf, sheaf elevation, angle of sheaf's major axis rotation from east, the caliber of the weapons firing, the number of weapons firing, and sheaf type.

Routine 'FA.ASSESS' is the major computational routine for damage assessment. It checks every target element on the battlefield (every red and blue vehicle/weapon system) to determine if that element lies within the sheaf of lethal effect. It does this by first making a rough check (i.e. comparing an element's distance from the aim point against a screening constant) to eliminate distant target elements that are obviously not endangered. It then makes a detailed check using a quadratic formula to determine if a near element actually lies within the sheaf. Each element found to lie within the sheaf is identified and given a fractional value (from 0.0 at the boundary to 1.0 at the aim point) which corresponds to its relative closeness to the aimpoint. The effects of microterrain (simulated boulders, trees, etc.) in the vicinity of each target element are included in the computations above and may have the effect of reducing a target's vulnerability to indirect fires. Once specified targets elements are properly identified as vulnerable to the artillery fire, the routine checks for dismounted infantry targets and causes them to seek cover (routine 'FA.HIDE'). Next, a table look-up is directed for each potential victim and returns with a Pk (probability of catastrophic kill), Pf (probability of firepower kill), and Pm (probability of mobility kill). This table is user input and gives a damage probability as a function of firing weapon caliber, ammunition type, sheaf type, target system type, target position, and type kill. Routine 'ATRIT' is called

from the ground model to make the actual determination of damage given the probabilities above. A random number is drawn from a uniform distribution (0-1). This random number is compared with P_k . If P_k is greater than the random number a k-kill is assessed against that target element. If a k-kill is not assessed, m and f damage values are computed for the current artillery volley and aggregated with past mobility and fire power damage values resulting from earlier fires. If the combined P_m is less than or equal to the random number an m-kill is assessed. The same procedure is used for f-kill. If during this computation both an m-kill and f-kill are assessed, the target element is declared no longer active in the battle. NOTE: It is acknowledged that measuring damage as a decreasing function of distance from the center of an artillery sheaf is a rough estimate and other more precise methods exist. However, the algorithm is adequate to convey the synergistic effects of indirect fire in the battle and more efficient in terms of cpu time saved. If a specific study required more precision in this area a more sophisticated (and time consuming) method could be substituted.

Communications is universally acknowledged as the critical element for successful artillery fire support and is explicitly modeled in the FA module. Both Tacfire and FM radio voice nets may be played. Information which would in reality be sent over a radio link to generate a sequence of events is routed by the model to events 'COMMO.ATTEMPT' and 'COMRECD'. These events check the states of radio nets, impose suitable delay times on information flow, and activate the next proper event/routine in the sequence. The capability exists [5] also to integrate electronic warfare simulation into the module.

As can be seen from the brief summary above, the 'STAR' FA module is in a very dynamic state with great potential

for further enrichment. The basic design has been one which insures maximum flexibility in response to user input. More detailed definitions of the individual field artillery events and routines are found in Appendix A.

III. THE COPPERHEAD SYSTEM

The Copperhead Cannon Launched Guided Projectile is a new field artillery munition designed to enable indirect fire systems to destroy moving and hard point targets that in the past were unengageable or subject to suppression fires only. Physically, it is a 155mm projectile containing a semi-active laser homing device (seeker) set in a clear plastic nose cone. The body contains fins and wings which deploy in flight, providing the round with a limited maneuver capability. When a forward observer equipped with a laser designator illuminates an enemy vehicle within range, the energy reflected from the target can be detected by the Copperhead while in flight. Corrections sent from the seeker to the servo-mechanism control the wings and fins and cause the round to steer to the target and thereby provide a relatively high probability of single-shot hit. If design specifications are achieved Copperhead will provide a unique opportunity for the efficient attack of tanks, armored personnel carriers, and other armored vehicles with little change required in current doctrine or field artillery operational technique. (The rest of the chapter is a summary of system characteristics presented in TC-6-30-1.)

Though slightly heavier and longer than the standard 155mm HE projectile the Copperhead requires no special handling at the firing site. However there are significant differences in the behavior of a Copperhead as compared to conventional munitions once the round is fired. The Copperhead flies ballistically to a point on its downward trajectory where guidance and control assemblies activate. At this point the round can acquire reflected laser energy, lock onto the target being lased and begin maneuvering toward it. How-

ever, the ground surface area in which the round can maneuver is limited. The area of maneuverability of the Copperhead round is called a "footprint" (see figure 1). The size of the footprint depends on a number of variables such as cloud height, shape of trajectory (either ballistic or glide) and gun-aim point range. Obviously, the earlier in its flight the Copperhead acquires the laser pulse the greater its ability to maneuver. The footprint is determined with respect to the ballistic aimpoint, which is the point on the ground where the Copperhead round would impact if it did not maneuver. The farther a target is from the target location sent by the observer, the lower the probability of hitting it. At the outside limits of the footprint the probability of hit is greatly reduced.

In order for the Copperhead to begin its maneuver phase it must acquire a reflected laser pulse within its field of view. This pulse is generated by the forward observer on the ground using a ground/vehicle laser locator designator or G/VLLD. The G/VLLD is the primary source of laser designation for Copperhead. It provides the operator with extremely accurate range and vertical angle information and, when properly aimed, can place coded laser energy on stationary or moving targets. The reflected energy provides guidance information for terminal homing munitions, such as Hellfire, Copperhead, and other Air Force and Navy precision guided munitions.

Success of a Copperhead mission relies heavily on the effectiveness of the communications link between the FO and the FDC. The FO must put laser energy on the target continuously during the last 13 seconds of the Copperhead flight or the probability of hitting the target will be significantly reduced. This creates the need for the FDC to send a "designate" message to alert the FO of this requirement. Normally, if multiple rounds are to be fired a "designate"

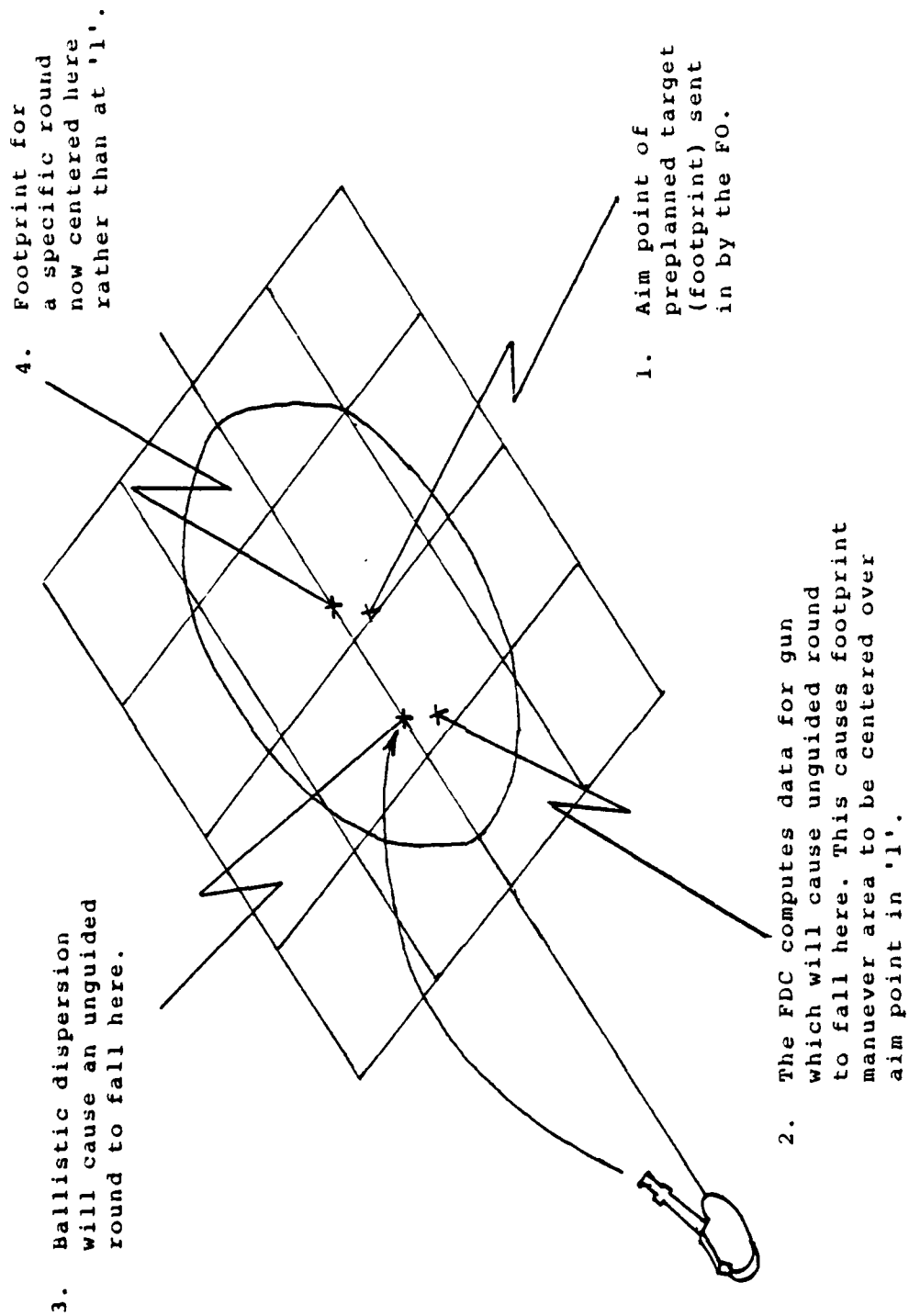


FIGURE 1. COPPERHEAD FOOTPRINT

message is sent only for the first round. The FO will lase for subsequent rounds at time intervals determined in the unit standard operating procedure. Precision and timing on behalf of the FDC to insure that the designator operator receives the correct commands, at the correct moment in time, are essential to the success of a mission. A delay of a few seconds or a lost radio transmission might well prevent the FO from properly lasing the target. On the other hand, commands sent too early might cause the FO to lase for too long a time thereby increasing his vulnerability to location and suppression by enemy elements equipped with laser warning devices.

When a Copperhead target is acquired, the request for fire will be transmitted over an established fire net to a firing battery FDC. The same battery that fires other close support field artillery missions for an observer will also normally fire Copperhead against targets of opportunity, and on-call planned targets. In the model all Copperhead missions will be fired for an observer by his Direct Support artillery battery.

The Copperhead-G/VLLD system should be treated as a direct fire weapon because of the G/VLLD's laser signature and the line-of-sight requirement of the G/VLLD-target link. For this reason the system must be included in the maneuver commander's fire distribution plan. Laser target designation requires that an uninterrupted line-of-sight exist between the designator and the target. Anything that obstructs the laser signal may become an unintentional target. Such interruptions include vegetation, terrain, dense smoke, dust, etc. The effects on observation resulting from general battlefield turbulence such as explosions, fires, dust, haze, etc, coupled with deliberate enemy employment of smoke, will certainly degrade copperhead employment. The engagement of moving targets with Copperhead can occur at

ranges that exceed those of other anti-tank systems, while preserving the basic load of ATGM's with frontline maneuver units. Copperhead-3/VLLD provides the capability to attrit the target array beyond the range of direct fire systems.

Though the Copperhead system has many advantages over those of conventional artillery there are a number of significant liabilities associated with the system. The table [3, p.7] on the following page identifies some key strengths and weaknesses.

TABLE 2: COPPERHEAD STRENGTHS AND WEAKNESSES

STRENGTHS

1. High hit probability on point targets, moving or stationary at ranges in excess of current direct fire weapons.
2. Highly lethal at all ranges.
3. Rapid rate of fire against an array of targets within the same footprint using volley fire.
4. G/VLLD operator and laser designator are easily concealed; designator does not have the pronounced firing signature of ATGM weapons.
5. portability of the G/VLLD
6. engagement with copperhead preserves the basic load of other direct fire weapons.
7. targets simple to acquire and track.

WEAKNESSES

1. Responsiveness of system is dependent upon several variables created by distinct acquisition and delivery components of the system.
2. G/VLLD operator and laser designator are vulnerable to suppressive fires
3. Copperhead system is dependent on two-way communications between designator operator and firing battery FDC.
4. Effectiveness of target engagement is limited by the observer's ability to track the target during last 13 seconds of the round's flight.
5. Emitted signal from the designator can be detected.

IV. MODEL DESIGN CONCEPT AND ASSUMPTIONS

The purpose of this chapter is to discuss the overall design concept and key assumptions used to integrate the Copperhead system into the STAR Field Artillery module. The modeling of COPPERHEAD, like that of other direct and indirect fire weapons systems, involves four significant operational phases: target detection, target selection, firing, and impact. The first three stages for a Copperhead mission are very similar to those of conventional artillery missions. For this reason the methodology for modeling the first three phases of Copperhead can be supported to a great extent by the logic currently installed in the FA module. The last phase, however, is the more difficult problem of modeling the processes involved with terminal homing and impact of the round- processes unique to Copperhead.

Figure 2 identifies the key events and routines of the FA module (shown in the rectangular blocks) which provide interface points to the COPPERHEAD-related routines and events (shown in the ovals). Routine UPDATE.CLUSTERS is the mechanism which generates all field artillery missions. In its modified state UPDATE.CLUSTERS will first check an FO's area of responsibility for a final-protective-fire mission. If none exists, routine COPPERHEAD will be called to check the FO's target list for potential COPPERHEAD missions. Such a mission is generated if, based on the current speed and direction vectors of the targets, more than a predetermined number will be within an active footprint for a specific time interval or 'band' which is associated with the system response time. If no mission exists, control is returned to UPDATE.CLUSTERS which will then sequentially check for a PRGR or a normal CLUSTER mission. This logic flow indicates

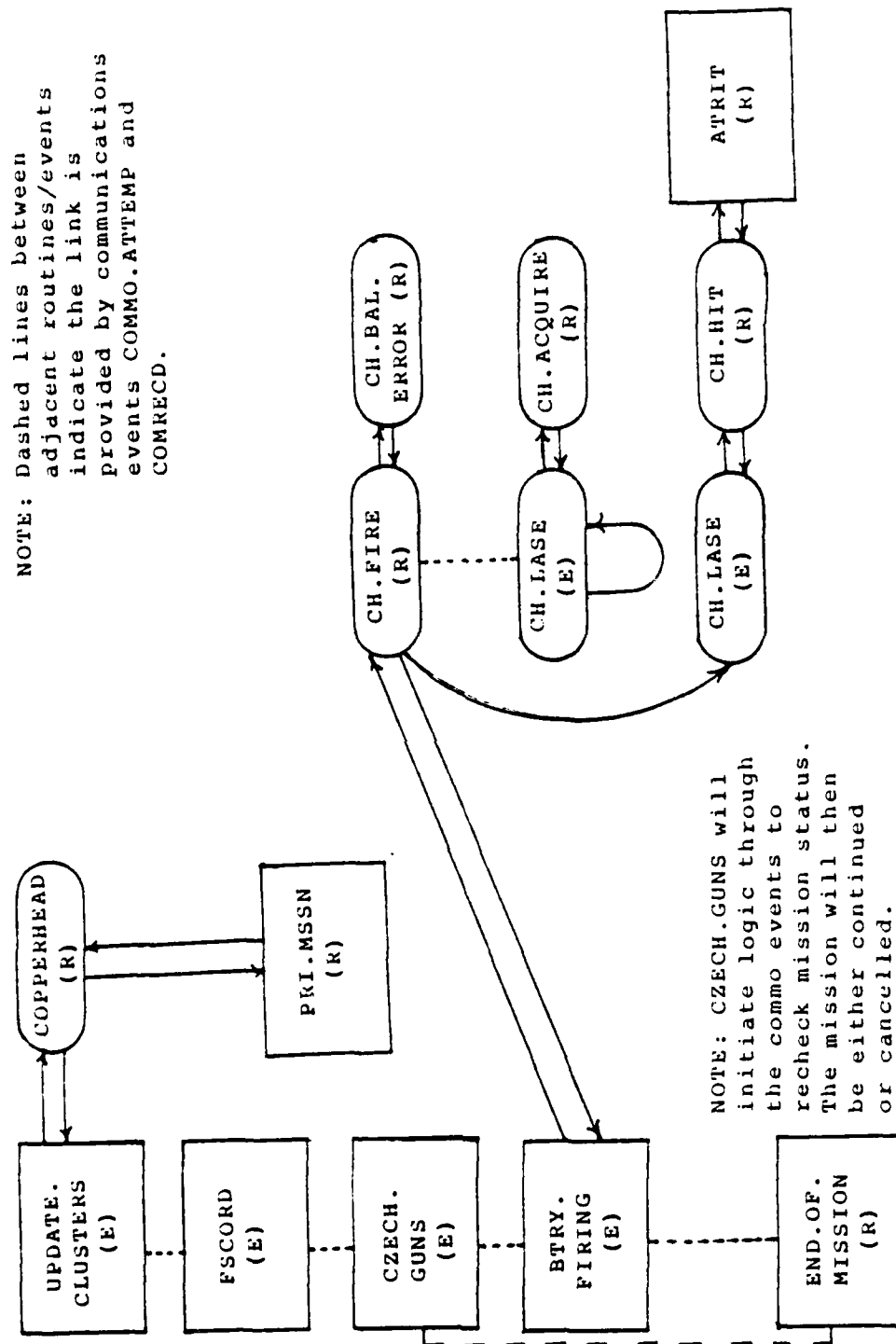


FIGURE 2. STAR-COPPERHEAD INTERFACE

that COPPERHEAD missions will be ranked just below the PPF in servicing priority. If routine COPPERHEAD does identify a potential mission it will call routine PRI.MSSN, which creates a MISSION. MISSION is a currently existing temporary entity which when modified will contain key attributes pertaining to COPPERHEAD -to include gun-aim point range, the coordinates of the aim point of the COPPERHEAD footprint, and the number of rounds selected by the FO to service the target array. If a COPPERHEAD mission is generated, control is again returned to UPDATE.CLUSTERS which in its last action will schedule the communication events COMMO.ATTEMP and COMRECD (not shown in figure 2). These two events will also be scheduled several times by events FSCORD and CZECH.GUNS. They will simulate actual radio traffic such as the FO's original call for fire, required intrabattery messages, a 'ready' message from the battery to the FO, and finally, the FO's 'fire' or 'cancel' message to the battery. In this respect, the model simulates a forward observer using the 'AT MY COMMAND' method as the method of fire control. Routine CZECH.GUNS will also select the gun from the observer's direct support battery to fire the mission.

If the FO sends the 'fire' message, routine BTRY.FIRING will be called, which identifies the mission as COPPERHEAD and calls CH.FIRE. It is in this routine where simulated firing of the projectile occurs. Here a temporary entity called ROUND is created and assigned a number of attributes such as time of firing and time of impact. Values for other key attributes, identified in the Preamble, will be assigned by later routines. An important point to consider when simulating the firing of a COPPERHEAD round is that, as in conventional munitions, ballistic dispersion will occur. The battery fire direction center provides to the gun specific data which is designed to center the round's maneuver footprint over the preplanned aim point selected by the forward

observer. Because of ballistic dispersion the theoretical impact point of an unguided round will be offset somewhat from the actual aim point. Routine CH.BAL.ERROR is called to address this fact. Also, in routine CH.FIRE the communications routines are employed to generate the 'designate' (turn on the laser) message from FDC to FO.

Event CH.LASE provides the framework to model the terminal homing processes that should start with approximately 13 seconds left in time of flight. This event selects a specific target from among possible candidates in the FO's target list, turns the laser on if appropriate, and calls routine CH.ACQUIRE. CH.ACQUIRE checks to determine if the round seeker has acquired the target and can begin its maneuver phase. The routine will be called repeatedly until acquisition is achieved or until simulation time equals time of impact. At the time the round is scheduled to impact routine CH.HIT will make a final determination of whether or not the COPPERHEAD round hit its target. If a hit is declared, a look-up table is consulted to provide damage probabilities. Specific parameters relating to such probabilities are passed to the STAR routine ATRIT which effects final disposition of the mission.

Two arrays must now be briefly explained in order to further clarify the general design of this model. As mentioned above, seeker acquisition is a key issue in the impact phase of the mission. It is important because the later that acquisition occurs within the final segment of flight, the smaller the maneuver footprint, which in turn reduces probability of hit. This fact is modeled in array R.HT.FP, which provides footprint parameters (lengths of semi-major and semi-minor axes of an approximating ellipse) as a function of gun-aim point range, time before impact, and altitude. The second array to be discussed is related to specific footprint locations on the battlefield. In STAR,

the user must determine a selected number of possible movement routes and positions for the units he is modeling. The program will determine which specific routes are taken and which specific positions are occupied based on stochastic interactions and internal decision logic. In the COPPERHEAD methodology, (which uses only preplanned targets) the user is required to develop the array AIM.PT which identifies footprint aim point coordinates for each possible battle position. The user must also specify whether these aim points will be priority or normal targets. In this manner, the FO will have a pre-selected number of valid Copperhead aim points available as pre-planned targets regardless of the position he was occupying. If the user believes that from certain positions no COPPERHEAD missions can be fired, that data can also be carried in the array AIM.PT.

Though an overview of the general modeling concept used in this thesis has been given in the paragraphs above, it is also important to explicitly define the assumptions made, since they form the foundation which supports the rest of the model. For this reason the following assumptions are identified and discussed.

1. The majority of Copperhead missions called in actual combat will consist of either planned priority or planned normal missions. (Copperhead target of opportunity missions are not played in the model.)

2. System degradation due to overspill/underspill is not significant.

3. The Lambertian reflectance distribution (cosine law) from the target closely approximates reflectivity values generated from a three dimensional target description.

4. As currently portrayed, the probability of hit is equal to the conditional probability of maneuver given acquisition. Probability of hit, probability of maneuver, and probability of acquisition are not determined in the

model from continuous distributions, but instead have zero/one values based on deterministic computations.

5. The use of elliptical footprints to approximate actual footprint geometry will not significantly impact on probability of hit.

6. The input data for footprint calculations, which was generated by other engineering models, is valid.

Some brief comments on these assumptions are in order and are provided below.

According to TC-6-30-1 [3, p.34], "The most desirable technique for the attack of both moving and stationary targets is the planned target technique. Normally the target of opportunity technique will be used only during mobile operations and during the time before development of planned targets." Assumption one simply identifies the planned target as the most significant engagement technique for COPPERHEAD. Later enrichment can incorporate the target of opportunity if desired. The second assumption refers to overspill and underspill of the laser energy that is being directed toward a target by the forward observer. There is a possibility that objects along the observer-target line other than the target itself may reflect some portion of laser energy creating false signals at the COPPERHEAD seeker. Attempts to model this phenomenon explicitly would be incompatible with the degree of resolution developed in STAR. (Such detail would be appropriate only in an engineering model). However, according to the PAM report [12, p.7], "The amount that a normally skillful operator at nominal ranges would degrade the system due to spillover/spillunder is not significant." The third assumption concerns the model's ability to determine probability of acquisition. Again, according to the PAM report [12, p.9], "While significant differences could exist in terms of the actual shape of the acquisition volume for

each reflected laser pulse, the spot jitter and time-variability of target heading is probably sufficient to smooth out the shape of the acquisition volume in such a way that the cosine law is approximately correct." The fourth assumption is closely associated with the two previous ones. The central idea here is, given that a Copperhead round has acquired a target and can physically maneuver to it, the probability of hit still remains some fractional value less than one. Unfortunately, the specific value of probability of hit is a complex function involving the sequencing and intensity of laser pulses arriving at the round seeker. In the short term, probability of hit has been set equal to probability of maneuver given acquisition. (See Chapter VII for further discussion.) Assumption five identifies another approximation technique used to enhance simplicity and minimize expense, which substitutes an ellipse of variable shape for actual footprint geometry. The footprint in reality is asymmetrical, but the use of an ellipse should preserve sufficient fidelity for a force-on-force model. The last assumption deals with a key issue in the overall modeling concept. Since COPPERHEAD's probability of hit is directly dependent on its associated footprint parameters, the validity of this model rests to a great extent on the validity of the footprint input data, which is not empirically based but comes from another model. J.A. Stockfish warns of this condition and states [14, p.vii], "One aspect of this situation is that the unverified findings of modeling conducted by one organization can be taken as fact by another organization and used as inputs for the latter's model." This case is certainly not an attempt to criticize the developers of the engineering model (ARADCOM and Martin-Marietta) but to point out to the decision maker that little empirical data exists on COPPERHEAD footprint parameters so further discretion should be used.

V. TERMINAL GUIDANCE AND IMPACT ROUTINES

As was mentioned in Chapter IV, a Copperhead mission cycles through four operational phases: target detection, target selection, firing, and impact. The current Field Artillery module already contains the required computer logic to support the detection phase of a COPPERHEAD mission. This phase has thus presented the least problem in developing. Target selection and actions involved with firing the round, however, require modification of code currently existing in the FA module and will therefore be discussed in the next chapter. This chapter deals with terminal guidance and the impact phase of a COPPERHEAD round- the modeling of which requires a completely new approach.

As was mentioned in the Introduction, the Army Material Systems Analysis Activity developed the primary algorithms used in this thesis to model the terminal effects of COPPERHEAD. Their approach- and the approach used here- is to divide the final trajectory segment into a set of two sequential deterministic checks. First, a check is made for seeker acquisition of reflected energy from the target. This reflected energy can be regarded as an acquisition energy volume emanating from the lased target. A precision guided munition intersecting this volume will immediately depart from its unguided trajectory and begin tracking. Given this has occurred, a second check is made to determine if the target being lased is within the physical limits of the round's maneuver area (footprint). The size of the footprint is directly related to the altitude in which acquisition is achieved. Figure 3 is a slightly modified version of the acquisition diagram found in the PAM report [12,p.11]. Given acquisition, the probability of hitting the target depends primarily on the distance of the target from the

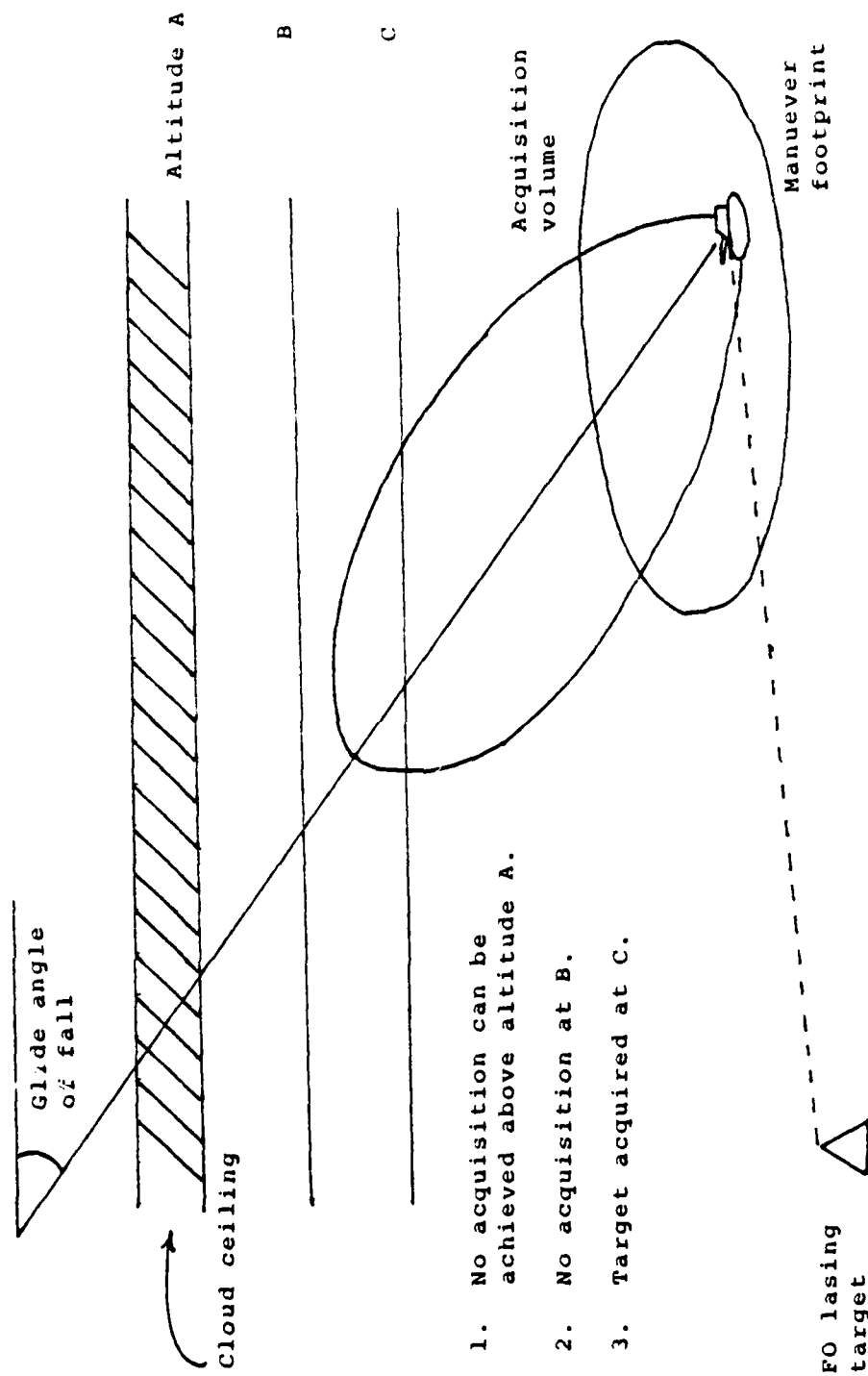


FIGURE 3. COPPERHEAD ACQUISITION

theoretical point of impact of an unguided projectile. Without acquisition the COPPERHEAD will follow an unguided ballistic or glide trajectory until impact with the earth.

A. CH.ACQUIRE

Routine CH.ACQUIRE contains the logic which conducts the acquisition check. The flow diagram is shown in figure 4. The first action of the routine is to compute the time remaining before round impact. This value, along with the gun-aim point range attribute of the 'MISSION', will enable the round's approximate altitude to be identified in the three dimensional look-up table 'R.HT.FP'. This table contains AMSAA provided input data displaying round altitude, angle of fall, and footprint characteristics as functions of gun-aim point range and time before impact. Once the appropriate data is located in the look-up table, the three dimensional position of the round at the current simulation time can be computed, as can the slant range to a specific target which has been identified earlier in the program. At this point the routine goes through some computations developed at AMSAA [12, pps.10-12] to determine the laser energy signal to threshold (S/T) ratio present at the round seeker. The equations are as follows:

$$S/T = (E_d * T_d * T_s * p * \cos \alpha) / (\pi * R_s * R_s * E_t)$$

where E_t = threshold energy density at seeker aperture
(joules/Km**2)

E_d = laser designator energy (joules)

T_d = designator to target transmission coefficient

T_s = target to seeker transmission coefficient

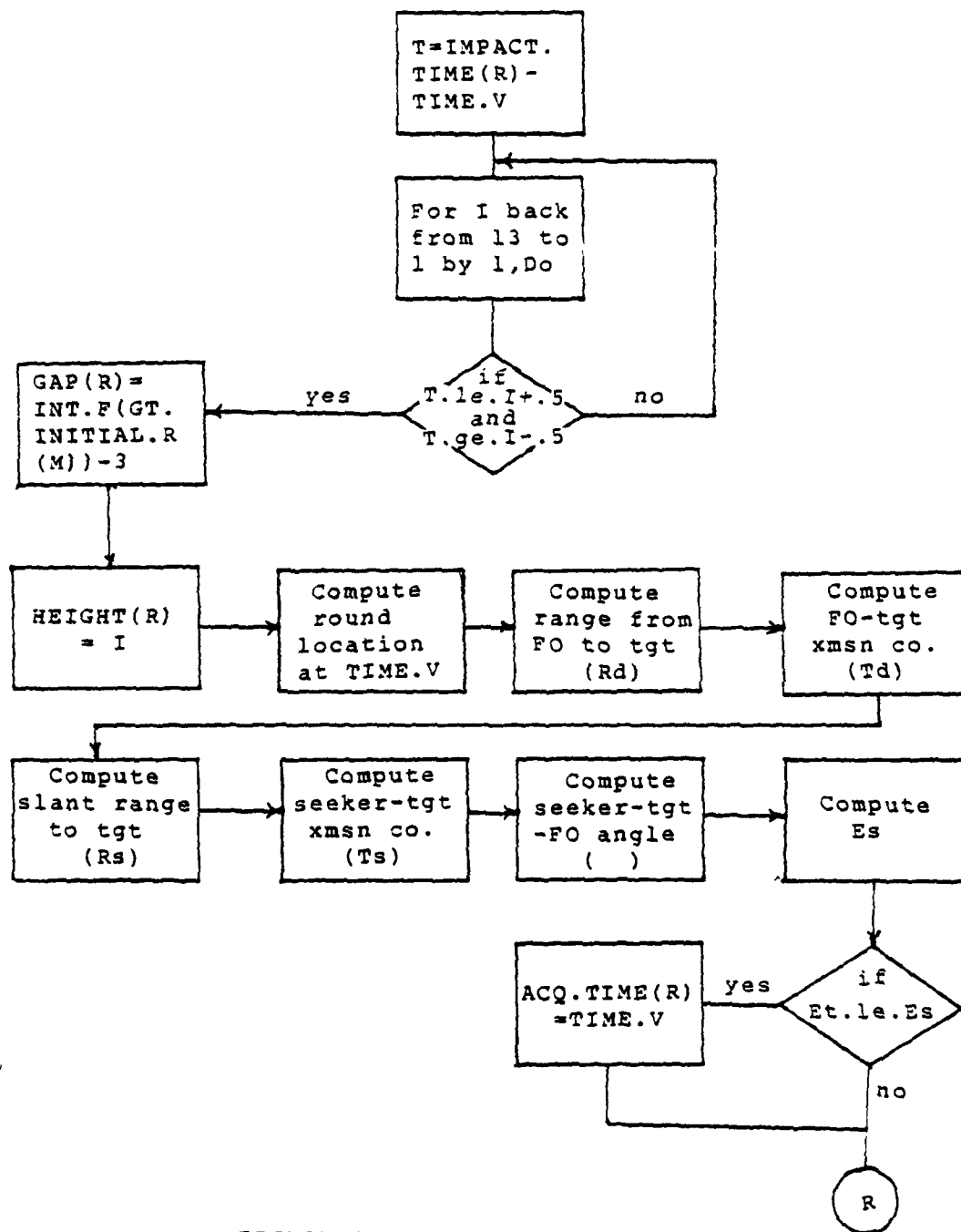


FIGURE 4. ROUTINE CH.ACQUIRE

p = target reflectivity
 α = lambertian angle (degrees)
 R_d = range from the FO to the target (Km)
 R_s = slant range to the target from the seeker (Km)
 GAM = atmospheric attenuation coefficient
 H_s = altitude at which the round first acquired the target (m)
 $T_d = e^{(-GAM * R_d)}$
 $T_s = e^{(-GAM * ((1 - e^{(.00025 * H_s)}) / (.00025 * H_s))}$
 $E_s = (E_d * T_d * T_s * p * \cos \alpha) / (\pi * R_s * R_s * E_t)$

Once these computations have been completed, a logic check is made to determine if the seeker's threshold energy density level is less than the density of the signal being received. If so, acquisition has been achieved and the current simulation time is placed in an attribute of the ROUND. If there is no acquisition the routine ends with no further action and must be called again at some small increment of time for a recheck of acquisition.

B. CH.HIT

Routine CH.HIT contains the logic which determines whether or not a Copperhead round, which has acquired a target, can physically maneuver to it. This is a yes/no decision made when the routine is called at the time of impact for a specific round. The flow diagram is shown in figure 5. The first action of the routine is to check whether acquisition has actually occurred. If not, the probability of hit for that round is zero and control is returned to the calling event, CH.LASE. If acquisition has occurred at some previous time, the routine determines if the target is within the area of the footprint associated with that acquisition time. It computes the difference in the x coordinate components and the y coordinate components of the target and the

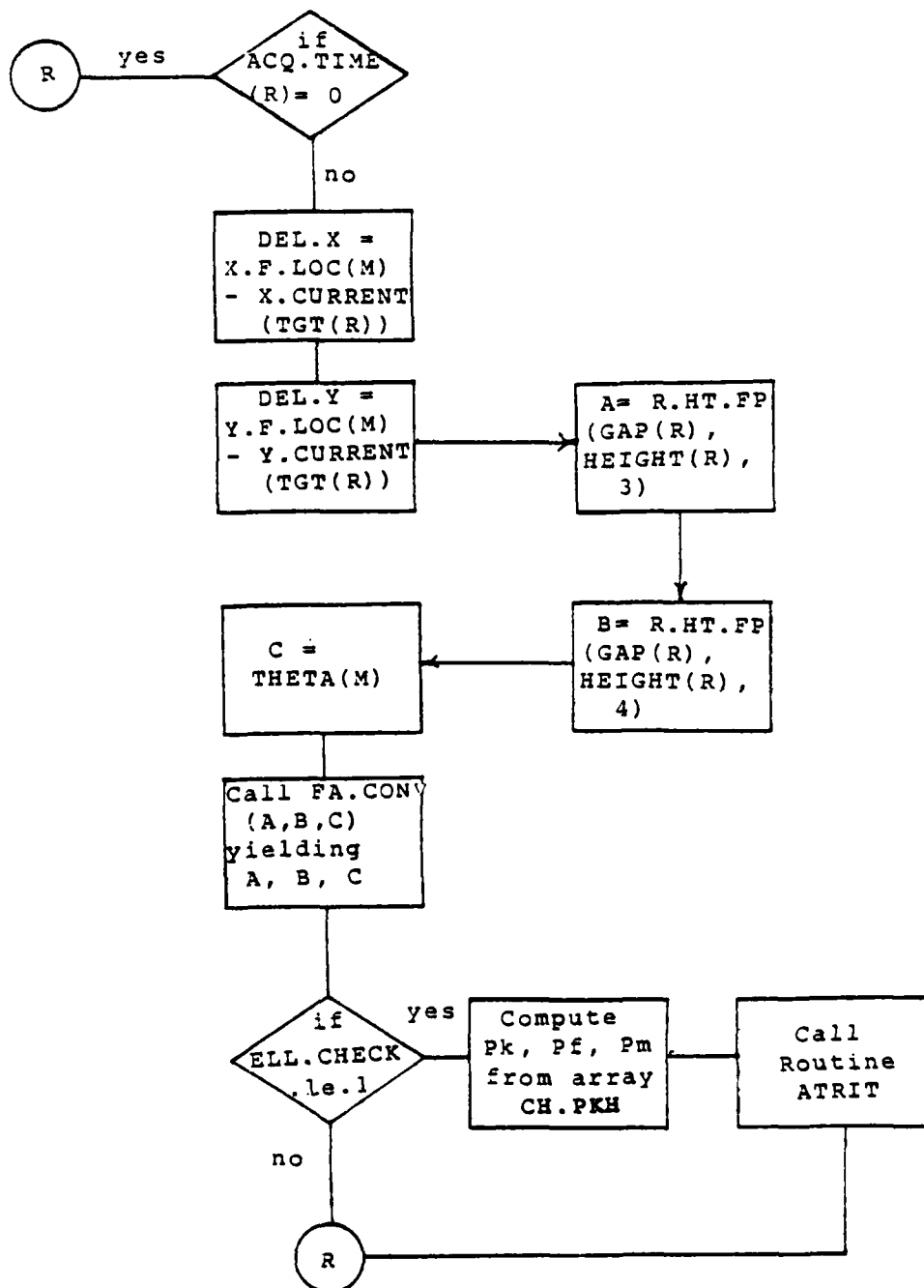


FIGURE 5. ROUTINE CH.HIT

theoretical impact point of the round. (Recall that the original coordinates of the preplanned aim point of the footprint have been modified in an earlier routine CH.BAL.ERROR to simulate round to round ballistic dispersion. Thus the theoretical impact point of the round is not the aim point but some offset coordinate). Next, the logic locates the length values for the semi-major and semi-minor axes of the footprint in array R.HT.FP. The gun-aim point angle is also identified. Using these parameters and routines FA.CONV and ELL.CHECK the logic mathematically centers an ellipse approximately equal in shape to the footprint over the theoretical unguided point of impact. If the target is within the ellipse the conditional probability of maneuver given acquisition is set equal to one. This implies that the probability of hit is also equal to one- thus a hit is scored on the target. (The calculation of probability of hit using this yes/no method is a rough approximation since, in reality, the round could miss an acquired target even if that target were within the specific maneuver footprint. Also, probability of hit monotonically decreases as target distance from the unguided aim point increases. Probability of hit obviously does not abruptly change from one to zero at some elliptical boundary. See Assumption 4, Chapter IV). If the target is not within the ellipse, the probability of hit is set to zero and control is returned to event CH.LASE. If a hit is assessed, the routine continues by searching the array CH.PKH which is an AMSAA provided look-up table containing the conditional probabilities of mobility kill, firepower kill and catastrophic kill given a hit. These probabilities are a function of the specific type of target (e.g. T-72 tank, BMP, ZSU-23, etc.) and the target's defilade status (fully exposed or in hull defilade). Once the values for the mobility kill, firepower kill, and catastrophic kill have been located, Routine ATRIT is called

and stochastically computes the final results of the engagement. ATRIT performs the same function here as in the conventional artillery mission (see Chapter II). If the engaged target is declared a catastrophic kill routine ATRIT also generates the logic which will remove it from the FO's target list. Such a target will not be selected again for further engagement by subsequent rounds in a multiple round mission. Targets which, however, were adjudged firepower or mobility kills can be engaged again.

C. CH.LASE

The event CH.LASE is designed to model the actions of the Forward Observer from the time he first attempts to select a specific target and begins lasing until the time of actual round impact. The flow diagram is shown in figure 6, at the end of the chapter. Ideally (in the real world and the model) the FO should receive a 'designate' message from his FDC twenty-two seconds before impact of the mission's first round. This message alerts the FO of the need to select a target and begin lasing in nine seconds (13 seconds before round impact). However, unusually long message response times or enemy jamming may prevent the message from reaching the FO at the appointed time. The first action initiated by event CH.LASE is to check the current simulation time against the time of impact of the round. If the times are equal the round has impacted. Routine CH.HIT is called and final processing is completed. If the current simulation time is earlier than round impact time, a check of the FO's status is made. If the FO has been killed, the acquisition time is set to zero. This will ultimately result in the round being regarded as a miss in the current model. (Suppression effects of rounds missing a target and damage effects of 'lost' rounds can be addressed later in an enrichment phase.) If the current time is less than time of

impact and the FO is still alive, the logic moves to one of two major branches controlled by the "IF LASER.ON(R) =ON" decision diamond. (LASER.ON is an attribute of ROUND.) If logic flow is sent to the left (no) branch it indicates that this is the first iteration of the event for a specific round and the laser has not yet been activated. If this is the case, the logic next attempts to select the best target from all possible targets currently within the footprint. (This footprint represents a rough estimate of the round's maximum maneuver area and corresponds to the template generated footprint used by the FO. It is not a specific footprint located in array R.HT.FP.). The best target in this case is defined as the target closest to the aim point at the current simulation time. A subroutine called BEST.TARGET has been designed to conduct this selection process. Basically, it identifies all targets within some user input distance from the aim point, screens those to which the FO does not have line of sight, and ranks the remainder in an array according to their nearness to the aim point. Its last action is to set a pointer to the Ith closest target to the aim point. The I value is carried in the third argument of the subroutine and will initially be set at 1.

Once the best target has been selected, a series of line of sight checks are conducted to that target's projected location in one second increments up to the time remaining until impact. This logic is established to prevent a 'best target' being selected which, though nearest the aim point, will apparently go into a defilade position at some time prior to round impact. Theoretically this will not happen often if the user selects aim points which are centered in areas providing clear LOS from designated battle positions. If the LOS checks indicate that continuous lasing of the 'best target' will be possible through time of impact, then this target is formally identified as the target of the cur-

rent round. The identification number of this target is given to TGT.ID- an attribute of ROUND. Next, the FO's laser attribute is set equal to 'on', a number of other key attributes are set to appropriate values, and routine CH.ACQUIRE is called to check for possible seeker acquisition. Finally, regardless of whether or not acquisition is achieved, a CH.LASE is rescheduled in one second.

If routine CH.LASE is initiated and the logic flow is diverted into the right (yes) branch of the "LASER.ON(R)=ON" decision diamond, a target for the round has already been selected and the FO's laser has been activated. The first action within this branch is to check for previous acquisition. If the round has not yet acquired, routine CH.ACQUIRE is again called and CH.LASE again rescheduled. If the round has in fact already acquired the target at some earlier simulation time, a current check for line of sight must be made. The reader will remember that a series of LOS checks were made in the left branch logic when the target for the round was selected. These checks were made to predicted future positions of the target based on the apparent speed of the target and its apparent direction of movement. The single LOS check made here in the right logic branch is not to a predicted point but rather is an actual check between the FO and the target in its present position. This is to insure that during the illumination phase the target does not suddenly change direction and move into a defilade position without a corresponding calculation by the logic to decrease probability of hit. If LOS has been lost the logic must simulate an attempt by the FO to transfer the mission to a new target. The modeling concept here is to search the forward observer's target list again for a new target to replace the one to which line of sight has been lost. This new target will not be selected based on its relative proximity to the aim point, but to the 'old' target. This is

done because when the old target was initially illuminated by the GVLLD, the round's flight path was diverted from the aim point as it began tracking. Once the new target is selected, the acquisition process must be reinitiated. For both CH.ACQUIRE and CH.HIT, however, calculations must be based on the old target location as the aim point rather than the original adjusted aim point. (This method is a crude way of addressing this complex problem of target switching. A more precise approach needs to be developed). If line of sight is still present between the FO and target, event CH.LASE is rescheduled.

Final processing of the round's effects is activated when event CH.LASE is scheduled at time of impact. Routine CH.HIT is called to determine if the round is scored as a hit or miss, and to make damage assessment if necessary. If this is the last round of the mission the ROUND entity is destroyed. If there are additional rounds remaining in the mission, the current ROUND is again destroyed, but a CH.LASE for the next ROUND is scheduled in 7 seconds. The selection of this time value of 7 seconds assumes that rounds are fired at twenty second intervals by the battery.

The event CH.LASE and the routines CH.ACQUIRE and CH.HIT constitute the essential logic of the impact phase of a Copperhead mission. The next chapter discusses the routines which pertain to mission selection and actual firing processes within the battery.

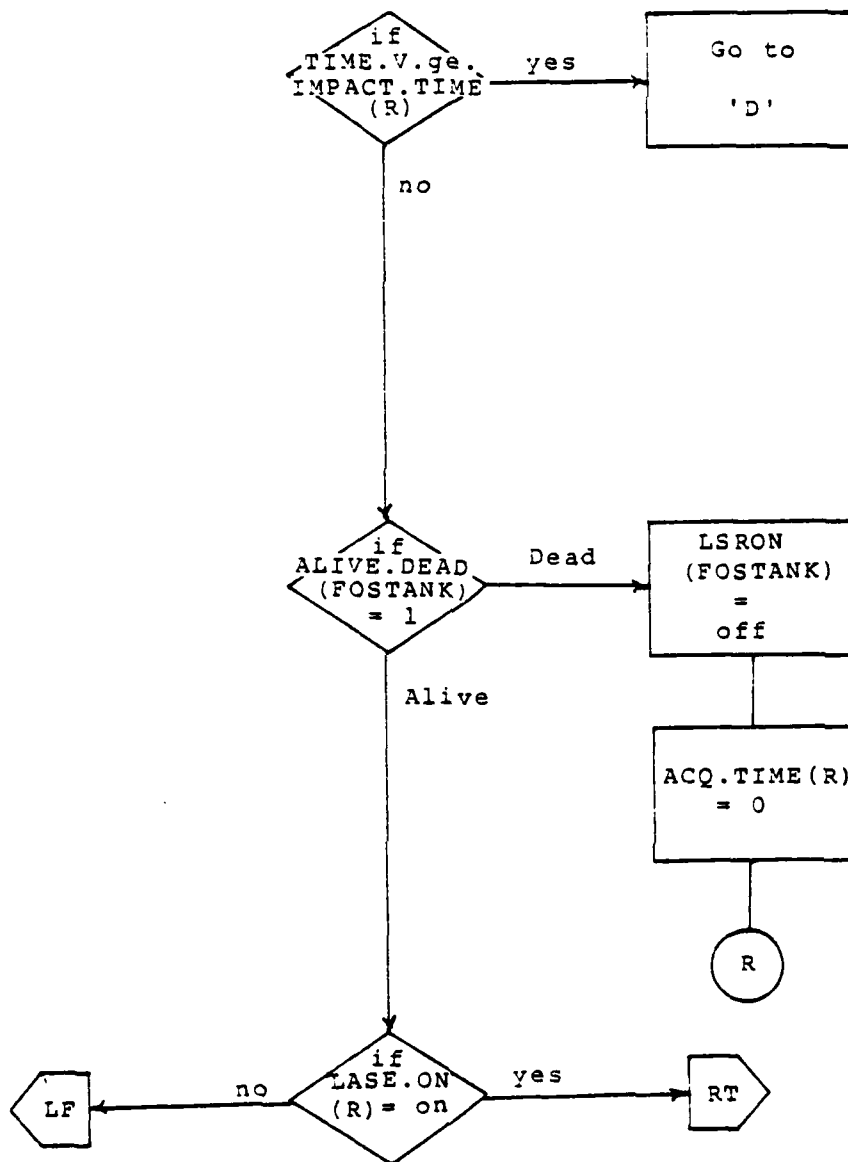


Figure 6. EVENT CH.LASE

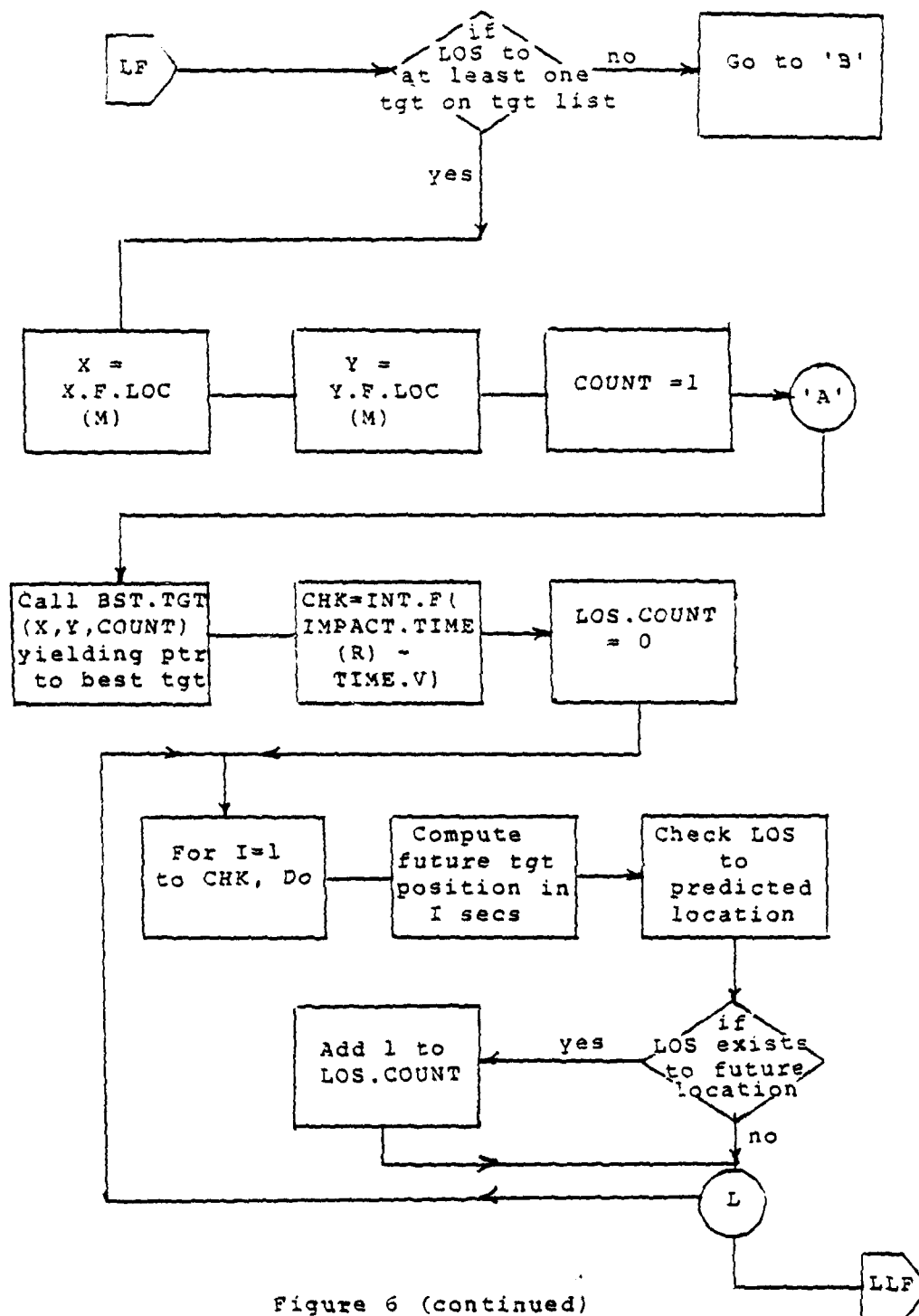


Figure 6 (continued)

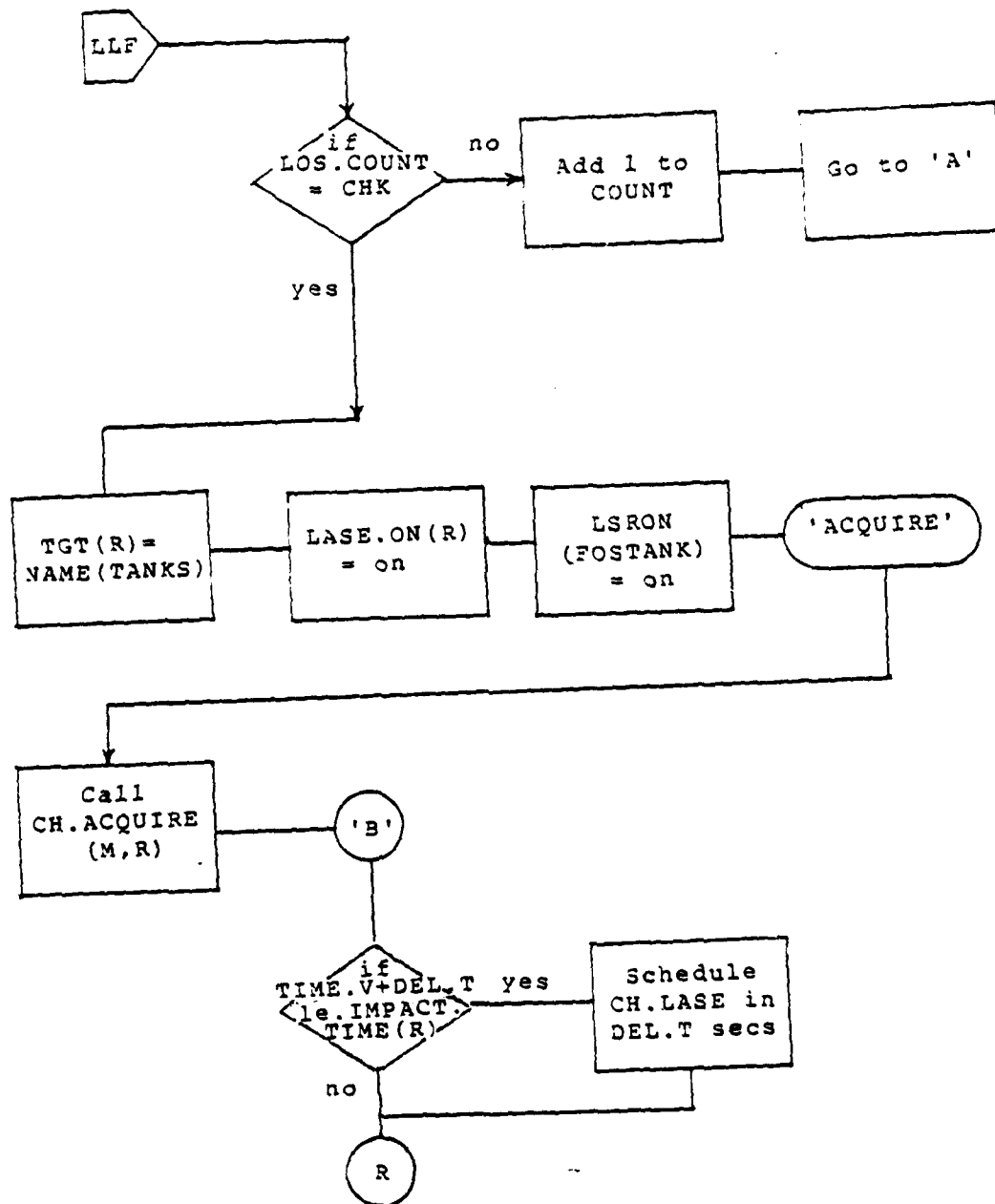


Figure 6 (continued)

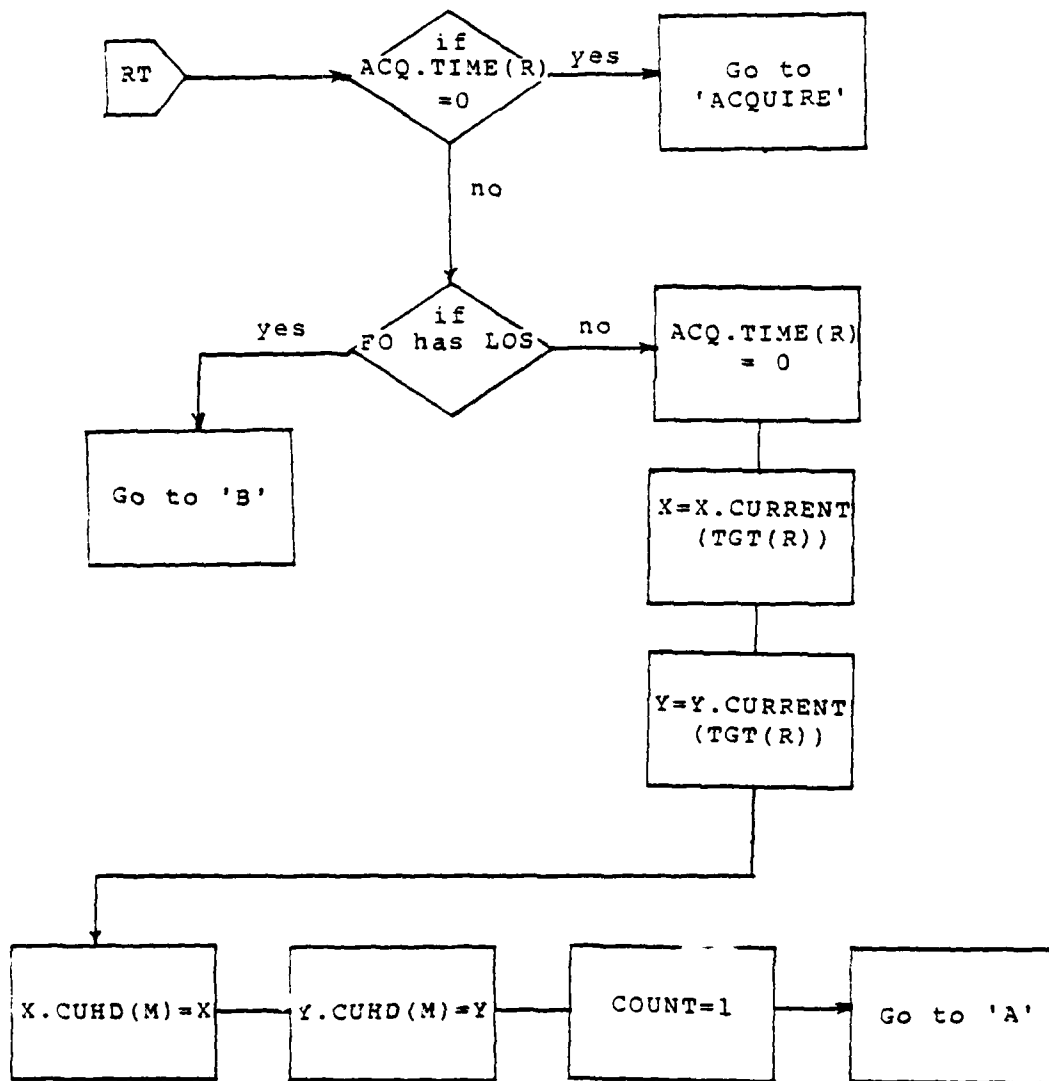


Figure 6 (continued)

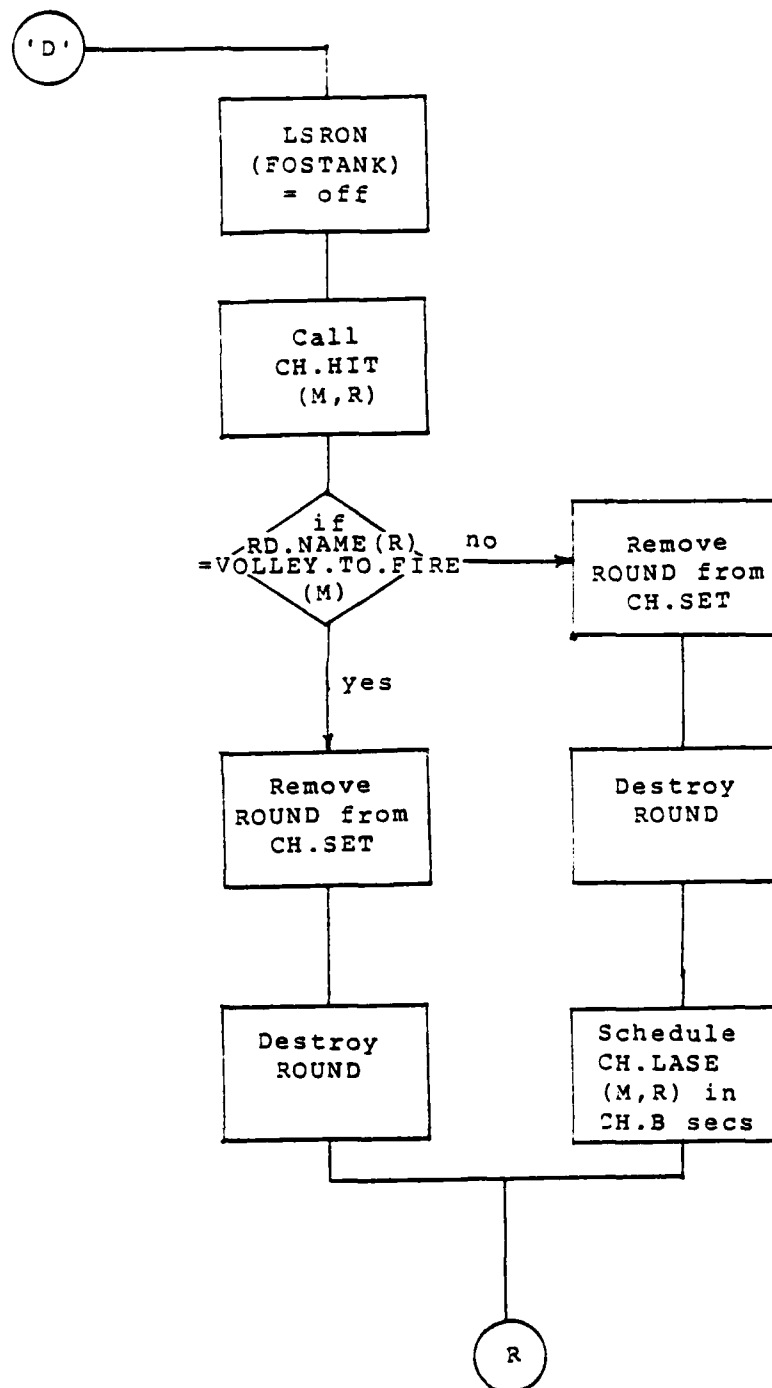


Figure 6 (continued)

VI. INTERFACE EVENTS AND ROUTINES

In the design discussion of Chapter IV it is mentioned that while the impact methodology of a Copperhead mission is unique, the other three phases have many modeling similarities with conventional artillery missions. In fact, the target identification phase of Copperhead, in which the forward observer identifies enemy vehicles and compiles a target list, is exactly the same as currently modeled for conventional artillery. Therefore this aspect of the Copperhead mission requires no modifications to current STAR logic. The target selection and firing phases, however, do require significant additions to the existing code in order to include Copperhead in the force-on-force simulation. Of the approximately 40 events and routines which comprise the Field Artillery module, six are identified as requiring modifications in order to support Copperhead. These are events UPDATE.CLUSTERS, FSCORD, CZECH.GUNS, BTRY.FIRING, COMRECD, and routine PRI.MSSN. The specific modifications required will be discussed in detail in this chapter. First, however, it is appropriate to identify the three remaining Copperhead routines which will provide the bulk of the logic modifications to be integrated into the above. They are routines COPPERHEAD, CH.FIRE, and CH.BAL.ERROR.

As was previously stated, event UPDATE.CLUSTERS is the segment of the Field Artillery module in which conventional fire missions are generated. This event checks sequentially within a forward observer's area of responsibility for conditions which meet the criteria of a final protective fire mission, a trigger mission, or a standard mission. With Copperhead integrated into the model, event UPDATE.CLUSTERS will first check battlefield conditions for a final protective fire requirement. If there are none, it will next call

routine COPPERHEAD, which determines if the Copperhead system should be employed.

A. COPPERHEAD

This routine first determines if the conditions for a Copperhead mission exist, and if so, generates further required actions within the program. The flow diagram is shown in figure 7. The first step of the routine is to identify all priority Copperhead footprints associated with the position currently occupied by the forward observer. For each priority footprint so identified, the logic will predict the number of targets from the FO's target list which will be in that footprint for a certain time band associated with a priority system response. The user will input two parameters which define this time band. One parameter- the lower bound- will be the expected system response time for a priority mission based on relatively ideal conditions. The second parameter is the value for the upper bound of the time band. This upper bound value could be equated to a variance factor of the response time distribution and caused by unscheduled (but routinely occurring) delays within the system. Using the apparent speed and apparent direction of movement attributes associated with each target, the routine will predict if the target will be within the footprint at both the lower and upper bound values of the time band. All such targets will be multiplied by a user defined weighting value which will be a function of target type (tank, BMP, etc) and distance of the aim point from the forward observer. The resulting values will be summed to produce the total target weight predicted to be in the priority footprint during some probable response time. To generate a mission, a footprint must have a target weight at least as great as a user defined threshold value. If more than one priority footprint meets this criterion, the one with the

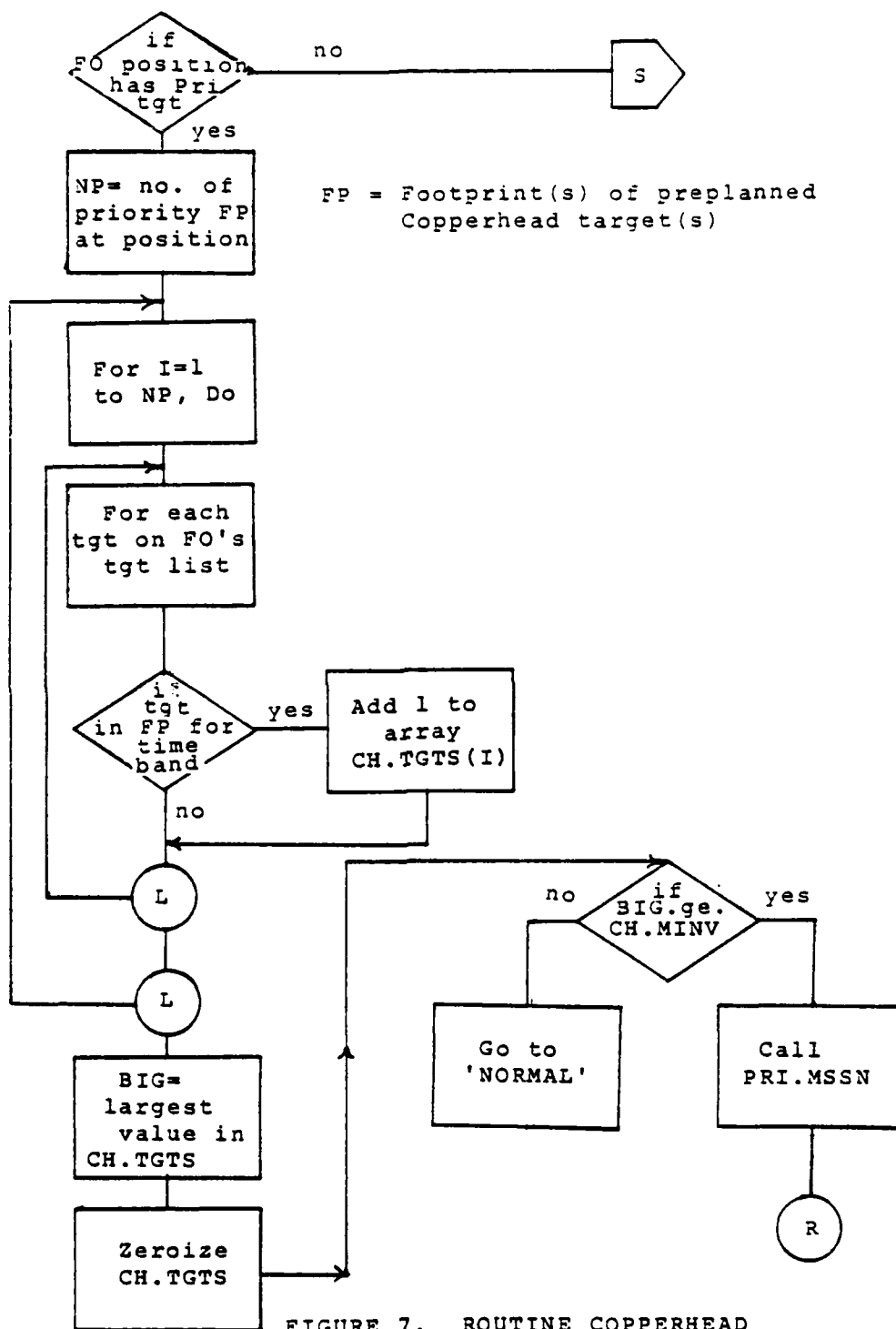


FIGURE 7. ROUTINE COPPERHEAD

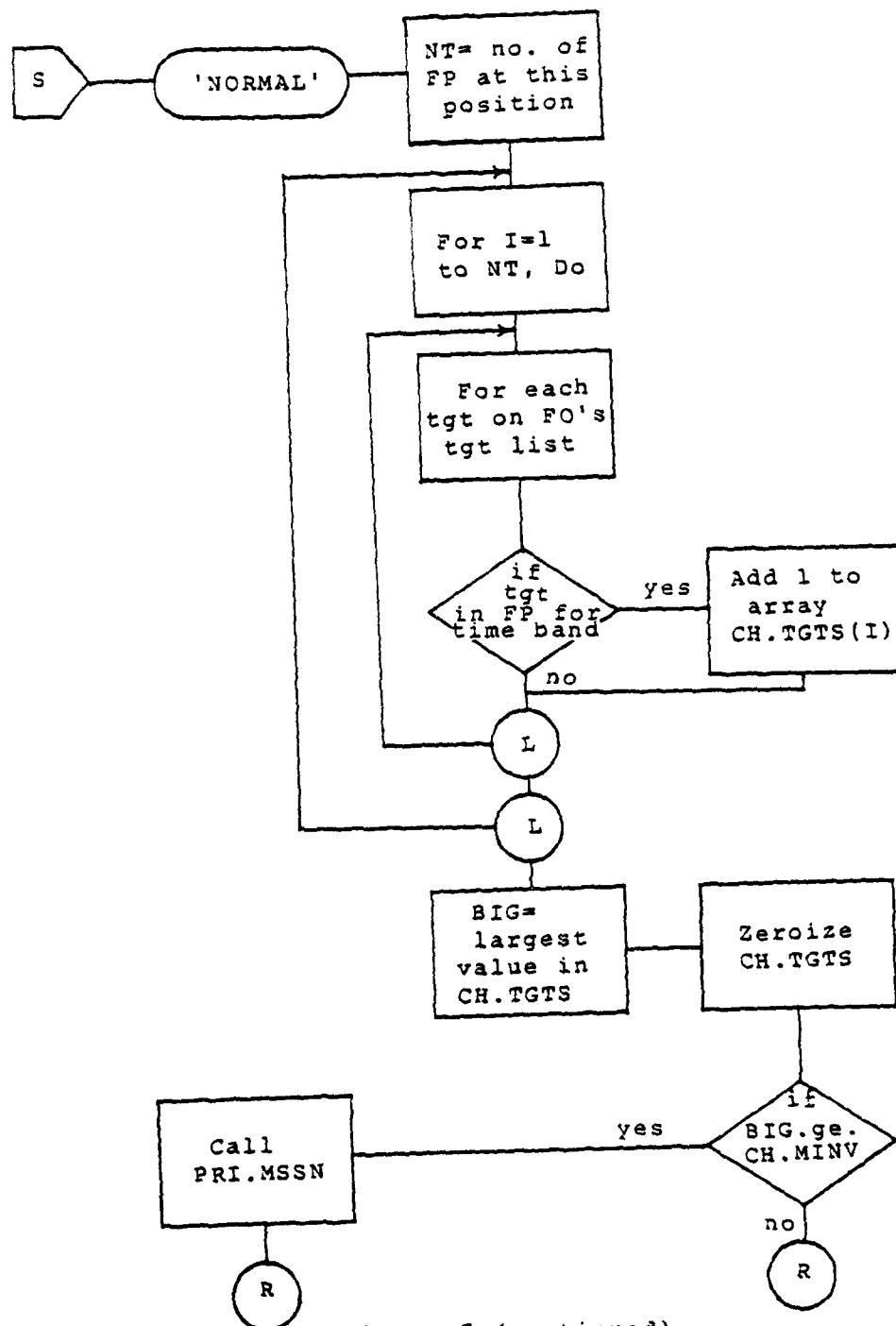


Figure 7 (continued)

largest target weight will be selected for the mission. If the position currently occupied by the FO does not have any associated priority Copperhead targets, the routine will perform in a manner similar to that discussed above, except that the parameters defining the time band will pertain to normal preplanned targets rather than priority targets. In either case, regardless of the type of footprint which has been selected, the routine PRI.MSSN is called to create a temporary entity- MISSION. It is also in PRI.MSSN that the FO's direct support battery is identified and most MISSION attributes are assigned a value. These attributes include AMMUNITION.TYPE, X.FUTURE.LOC, Y.FUTURE.LOC, GT.INITIAL.RG, THETA, etc. (see Appendix C).

Once a Copperhead mission has been generated and control is returned to UPDATE.CLUSTERS the final action to take place in this event is the scheduling of a COMMO.ATTEMPT. This begins the message play which is a critical factor in the modeling of Copperhead. A brief digression is appropriate here to provide a general concept of how radio communications are accomplished.

Communications is explicitly modeled in STAR primarily through the interaction of two events - COMMO.ATTEMPT and COMRECD. Before these events can be employed, however, a one dimensional array ,MEMB, must be created in whatever event or routine initiates the message. MEMB(1) is a six digit number which identifies the sender of the message, its intended recipient, and the type information to be transmitted (see Appendix A). The rest of the MEMB array consists primarily of pointers which identify specific entities. (Remember that there may be a large number of forward observers, rounds, batteries, and missions in existence at any one time in the simulation. Pointers provide the mechanism which links the proper set of permanent and temporary entities together). Once MEMB has been filled with

appropriate values, event COMMO.ATTEMPT is scheduled. This event portrays the technical processes that take place in a tactical single channel radio net when one station attempts to contact another. The event makes a number of checks of the system to include: a check of electrical line of sight, a comparison of distance between stations with the radio planning range, a check to insure both stations are in the same net and a check to insure the distant station is monitoring the net. The results of these checks come from either deterministic or stochastic computations, but regardless of a specific outcome, a certain finite period of time is calculated to have occurred during the communications attempt. If no contact is made, the calling party will make additional attempts until some user cutoff time. If contact is made, event COMRECD is scheduled. COMRECD is the second key event in the communications process. It triggers the response actions of a called party resulting from the information provided by the calling party. MEMB(1) routes the received message to the place in the event which contains the appropriate decision logic for the specific message. Table 3 displays the MEMB(1) values, message descriptions, and responses to messages relating to the Copperhead mission. Attention is now returned to the original interface discussion.

As was mentioned in a preceeding paragraph the last action of event UPDATE.CLUSTERS is to schedule a COMMO.ATTEMPT. This event - after imposing a suitable delay time- carries a Copperhead fire mission request to COMRECD which in turn schedules a FSCORD at the current simulation time. Event FSCORD performs two key Copperhead functions. First, it sets a flag which prevents event UPDATE.CLUSTERS from calling routine COPPERHEAD for any FO who is currently engaged in an active Copperhead mission. This avoids the possibility of the program creating a second simultaneous

TABLE 3: COPPERHEAD MESSAGE TRAFFIC

<u>MEMB(1)</u>	<u>MSG</u>	<u>RESPONSE</u>
010213	Copperhead fire mission	Schedule a FSCORD now.
020313	Copperhead fire mission	Schedule a CZECH.GUNS
020119	Gun ready	1. Determine if sufficient targets will be in the footprint in round time of flight. 2. If 1. above is yes send a 'fire' message (010220). 3. If 1. above is no send a 'cancel' message (010203).
010220	Fire	Schedule a BTRY.FIRING now.
010203	Cancel	1. Send a message to the FSCORD cancelling the mission. 2. Schedule an END.OF.MISSION now.
020303	Cancel	Release gun firing Copperhead mission back to the direct support battery.
020121	Designate (lase)	Schedule a CH.LASE at 'time of impact - 13' or now , whichever is later.
020114	Rounds complete	Schedule an END.OF.MISSION.

Copperhead mission for any one FO. Second, FSCORD prevents any direct support artillery battery from servicing more than two Copperhead missions at any one time. FSCORD's final action is to schedule a COMMO.ATTEMP which carries a Copperhead fire mission notice to COMRECD. Within COMRECD, this MEMB(1) value causes a CZECH.GUNS to be scheduled in an appropriate time interval.

Event CZECH.GUNS simulates the actions within a battery fire direction center upon receipt of a fire mission. For Copperhead the event will first determine if the mission is priority or normal. Next, the status of the guns will be checked and one specific gun will be selected to fire the mission. Values for array MEMB will be provided and a 'ready' message will be sent to the forward observer by scheduling event COMMO.ATTEMPT in an appropriate time interval. An appropriate interval in this case is a function of current gun status (idle or firing a mission) and urgency of Copperhead mission (priority or normal). The modeling concept here is to immediately accept any Copperhead mission regardless of the firing status of the battery. Even if the battery is engaged in a fire for effect phase or final protective fire, one gun will be directed to immediately prepare to fire the Copperhead mission. (More complex logic allowing the user to select gun allocation criteria can be developed as an enhancement.) Upon termination of CZECH.GUNS, the logic flow stops until COMRECD is processed for a 'ready' message from the battery. When this occurs, the program simulates the actions of an FO making a recheck of speed and direction vectors of the enemy vehicles on his target list. (NOTE: This segment of code could either be written into event COMRECD or, more probably, could be developed as a independent routine called by COMRECD. If such a separate routine were written, it could also be used to perform a similar function within event COPPERHEAD.) If

at this time the logic determines that sufficient vehicles will be within the maneuver footprint in time of flight of the round, it will take the normal actions to process a 'fire' message to be sent to the battery. If the direction of enemy vehicles has changed and the program determines that an insufficient number will be within the footprint, it will generate a 'cancel' message. In both cases the standard COMMO.ATTEMPT and COMRECD events will be scheduled. A MEMB(1) value identifying a 'cancel' message within event COMRECD will cause 'abort' messages to be generated, which in turn cause appropriate end of mission processes to occur. If a 'fire' message is input to event COMRECD, a BTRY.FIRING is scheduled at the current simulation time. The following major segment of Copperhead logic is called from event BTRY.FIRING.

B. CH.FIRE

This routine is the section of the program in which the firing of a Copperhead round is actually simulated. The diagram of logic flow is shown in figure 8. The first action of the routine is to create a temporary entity called ROUND. The next four logic blocks assign values to some of the attributes of the ROUND. In this four block segment, routine CH.BAL.ERROR is called which, when given the coordinates of the aimpoint of the preplanned footprint, returns with stochastically assigned delta x and delta y dispersion errors. Next, the ROUND is filed in a first in first out set. This will help to insure that after impact, the rounds will be destroyed in the proper sequence. The first decision diamond encountered in the logic determines if additional rounds are to be fired for this mission. If this is the last round to be fired, an attribute of the MISSION is set to a value which will initiate the required end of mission processes upon return of control to BTRY.FIRING. If there will

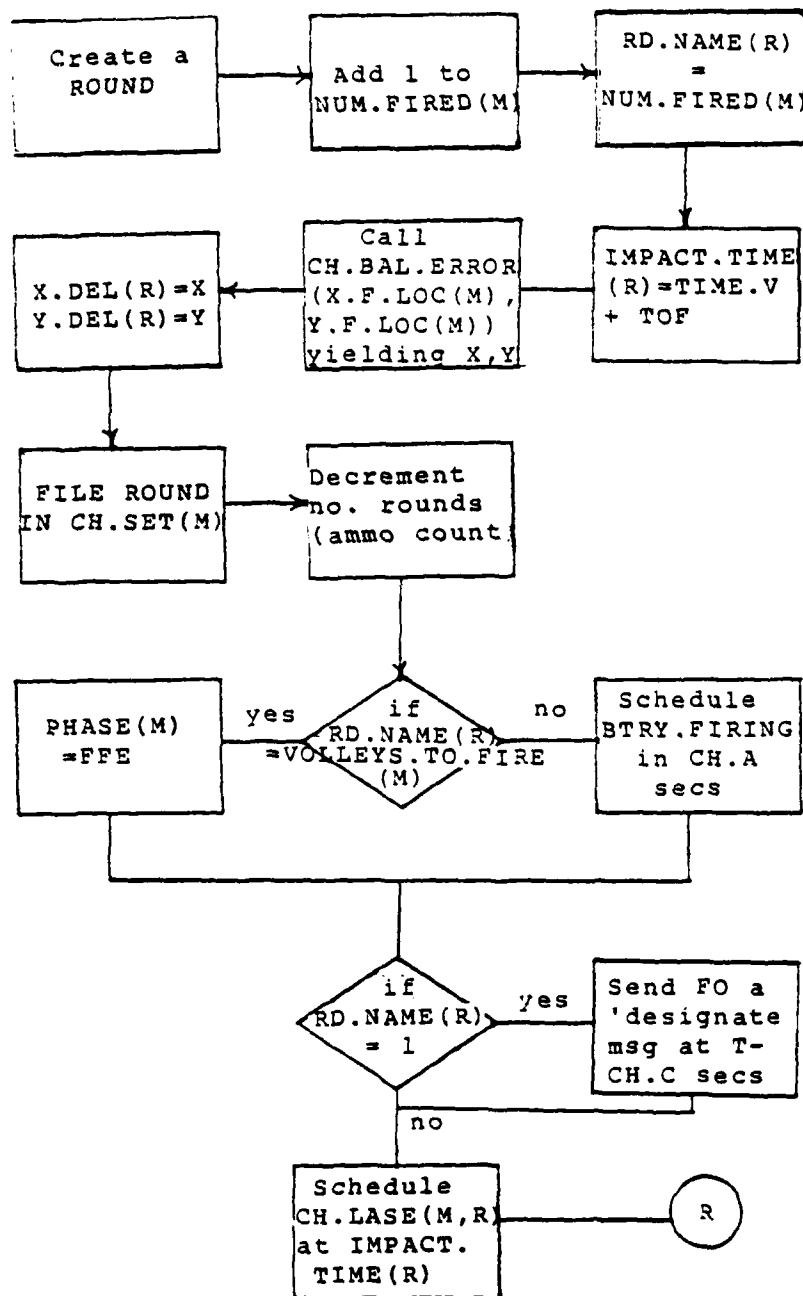


FIGURE 8. ROUTINE CH.FIRE

be subsequent rounds fired, a BTRY.FIRING is scheduled in twenty seconds. The second decision diamond determines if this is the first round to be fired in the mission. If so, a designate message must be sent to the FO at time of impact minus twenty-two seconds in order to alert him to begin lasing. The last action to be undertaken is a scheduling of event CH.LASE at the time of impact of the round. This insures that regardless of what happens to the designate message or to the FO, a final assessment of the effects of each round will be made by CH.HIT and that ROUND will be properly destroyed.

C. CH.BAL.ERROR

This routine allows the model to simulate Copperhead round to round dispersion caused by ballistic error. Equations must be computed to fit the curves shown in figure 9 [12, p.14]. For each round the gun-aim point range will be substituted for the equation's independent variable and a corresponding standard deviation would be computed for range/deflection error. From a $N(0,6)$ distribution, delta x and delta y would be determined probabilistically. An important point is that the delta values returned to routine CH.FIRE by this routine are with reference to the gun-aim point axis and not to the regular battlefield coordinate system. Routine CH.ACQUIRE uses these delta values in this format for its own internal computations.

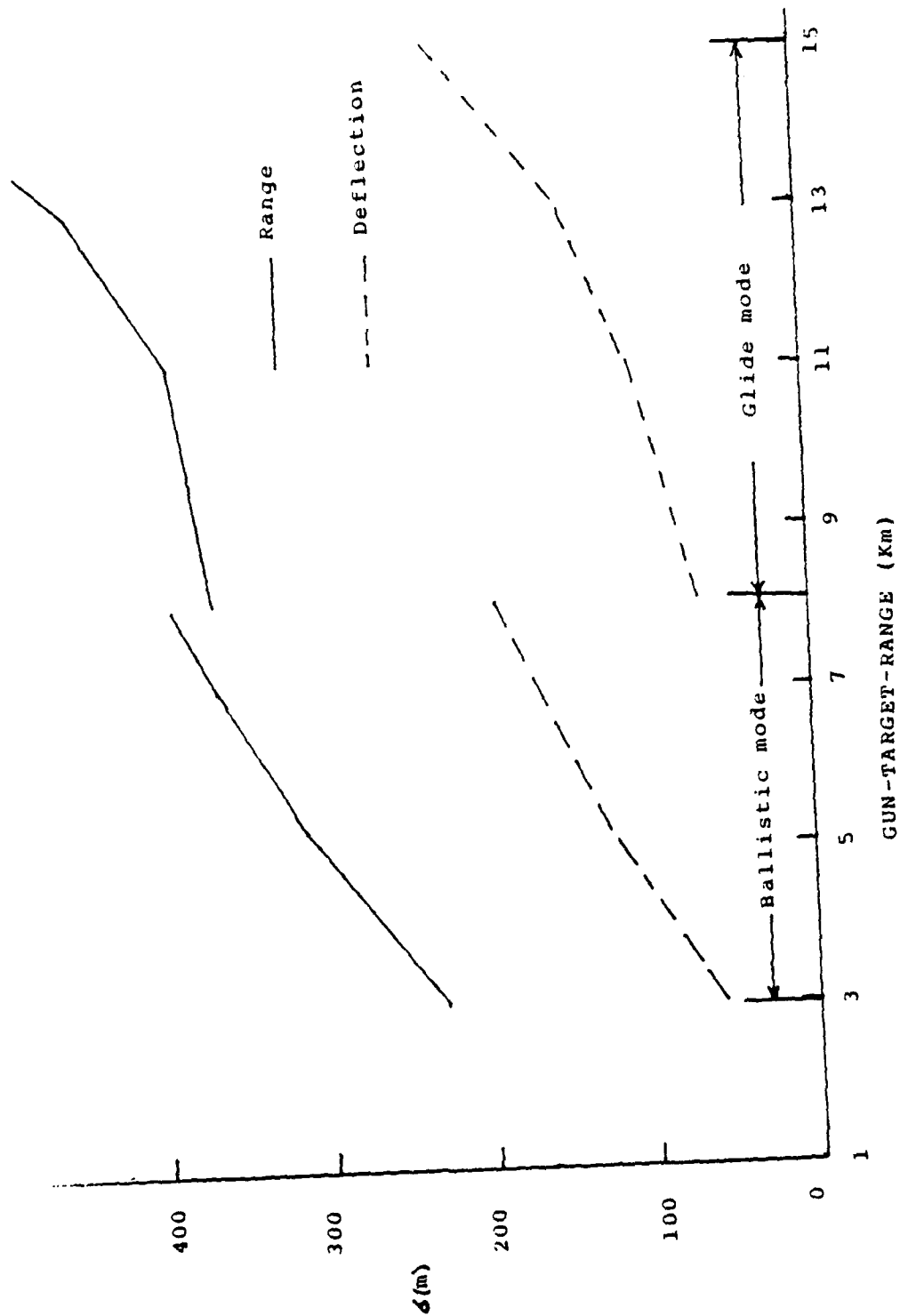


FIGURE 9. COPPERHEAD BALLISTIC ERROR

VII. SUMMARY AND FUTURE MODEL ENHANCEMENTS

The Copperhead modeling methodology established in this thesis provides the framework for the incorporation of this new artillery system into the STAR model. When coded, the logic presented will allow preplanned Copperhead missions to be played within the context of a force on force combat simulation. The logic design will permit a number of sensitivity analyses to be conducted which pertain specifically to Copperhead system performance. The impact on mission success of critical parameters such as cloud height, angle of fall, visibility, and system response time can be studied by changing selected input variables or arrays. More important, however, is the fact that synergistic effects of Copperhead as a combat force multiplier will now be available for evaluation.

The design approach developed in the thesis divides the Copperhead modeling logic into four segments corresponding to the following phases of a firing cycle: target identification, target selection, firing, and impact. The last segment of logic - Copperhead impact- is unique to this weapons system and is built around a 'probability of acquisition and maneuver' algorithm originally developed at the Army Materiel Systems Analysis Activity. Given the location of a round at some specific point along its trajectory the algorithm will first determine if acquisition (i.e. seeker lock-on of laser energy reflected from an illuminated target) is possible. If so, the algorithm will next determine if the target is within the maneuver area (footprint) of the round. If both determinations are positive, the model currently assigns a hit to that specific round ($P_{hit}=1.0$). Copperhead routines CH.ACQUIRE and CH.HIT accomplish these calculations. As will be discussed later in the chapter

there are a number of refinements needed in this probability of hit concept to portray the interaction of additional key parameters of the system such as spot jitter, false targets, etc. These refinements will increase the precision of the algorithm.

The first three logic segments of the Copperhead design have many similarities with the code already written for the conventional artillery systems in STAR. In fact, the target identification phase of a Copperhead mission is identical to that of a conventional mission and requires no logic modifications. The target identification and firing phases however, require significant additions be made to the Field Artillery module at a number of different interface points. Routines COPPERHEAD and CH.FIRE accomplish the bulk of the target selection and firing actions, but a number of other 'fixes' must be made - especially in the STAR communications events. Event CH.LASE is the key logic segment in the Copperhead model. It controls the integration of the initial detection phase, the firing phase, the communications exchanges, and the final impact phase.

Once the Copperhead logic has been coded and successfully integrated into the FA module, the last remaining requirement will be to validate the results. Unfortunately, little empirical data exists at the present time. Hopefully, as more instrumented field testing data becomes available, the logic and probability tables can be reexamined and adjusted to better represent the Copperhead system.

Though this thesis has attempted to provide a solid basis for STAR-Copperhead integration, there still exist some related areas requiring significant enrichment effort. The most pressing need, obviously, is to convert the logic diagrams into computer code. Once this is accomplished, at least five additional problems areas will need further attention. The first area concerns the necessity to include

the Copperhead target of opportunity as well as the preplanned mission, as a model capability. As currently portrayed, the model simulates preplanned Copperhead targets only, which limits the scope of possible combat scenarios. Professor J.K. Hartman's field routine approach 8 seems to offer some potential for solving this problem.

A second shortcoming of the current logic is the restriction that a forward observer can only obtain Copperhead fires from his direct support battery. The current Field Artillery module can simulate artillery units in the direct support, general support, and reinforcing roles, so an increase in the sources providing Copperhead fires must be made. A third area in need of further investigation concerns the analysis of suppression effects on an FO who is lasing a target. How will the probability of a target hit be degraded if an FO is subject to suppressive fires during the period he is designating a target? Suppression has always been a difficult phenomenon for combat modelers to define quantitatively. When suppression is played in conjunction with such a complex system as Copperhead, the credibility of any modeling algorithm will be extremely difficult to establish.

The last problem is especially critical to the effective modeling of Copperhead within a force-on-force simulation. This problem is the difficulty in constructing a probability of hit algorithm which can strike a balance between effective representation of critical system parameters and design resolution. On one hand, the modeler wants to portray real world effects such as spill-over and spill-under, false laser-target reflections, obscuration, laser spot jitter, target evasive maneuvers, etc, in an extremely detailed manner because these considerations are key in measuring Copperhead performance. On the other hand, however, a point is reached where the addition of more performance characteristics into

an increasingly complex engineering algorithm does not provide a corresponding return in effectiveness of the overall model. This is especially true since a data base does not exist against which an algorithm can be validated. The status of the current probability of hit algorithm (CH.ACQUIRE, CH.HIT) has not yet established this balance. It appears that more resolution can be gained. Some of the above listed effects/parameters are either ignored or 'assumed' away in the model. As was discussed earlier, the solution to the problem would not be attained by trying to install a 'engineering' segment into the program. However, it appears that much success can be achieved by utilizing a current 'engineering' model such as LDWSS [10] to provide a probability of hit look-up table. This table would reflect probability of hit (given acquisition and maneuver) as a function of those critical parameters mentioned above. To determine if a specific round hit a specific target, acquisition and maneuver would first be checked. If the determinations were positive, the look-up table would be accessed, the probability of hit located, and a stochastic draw made to score the round.

APPENDIX A CURRENT STAR FIELD ARTILLERY EVENTS AND ROUTINES

ALT.FO

This routine switches the forward observer's function to a platoon leader or a maneuver company commander if the FO is 'killed'.

BTRY.FIRING

This event computes the center of impact of a volley of artillery fire and then schedules an 'INCOMING' in time equal to time of flight of the artillery projectiles. After the rounds are fired, logic is checked to determine if any new FPF or TRGR missions (priority) are waiting in queue. If so, a message is sent to the 'FSCORD' advising that the present mission is being preempted. If the current mission is not bumped by a higher priority mission: 1. further adjustment is required or 2. a FFE is still in progress and 'BTRY.FIRING' is rescheduled or 3. the last volley of FFE has been fired and a 'rounds complete' message is sent to the Fscord.

CZECH.GUNS

This event processes fire missions when they arrive at the battery. If the battery is idle the mission, regardless of type, is immediately sent to the guns by scheduling BTRY.FIRING. If the arriving mission is an FPF or Trigger mission, the battery between adjusting rounds, and the current mission a standard target of opportunity, the current mission is preempted in favor of the higher priority mission. If the battery is busy with a mission when a new mission of lower priority comes in, the new mission is placed in the howitzer queue to await later processing.

COMRECD

This event terminates a successful communications attempt by initiating the logic flow corresponding to a normal response by the called party. MEMB is a one dimensional array whose location is identified by the single argument of event COMRECD. MEMB(1) is a 6 digit number (1st 2 digits- origin of msg, 2nd 2 digits- destination of msg, 3rd 2 digits- msg type). MEMB(2)- MEMB(n) contain other data, such as pointer information, needed to pass the message to the right place. MEMB(1) contains the following information:

FROM (DIGIT 1,2)
 01 FO
 02 FSCORD
 03 BTRY PDC
 04 Q37
 05 Q36
 06 REINFORCING BATTALION

07 MANUEVER UNIT

TO (DIGIT 3,4)

01 FO

02 FSCORD

03 BTRY FDC

04 Q37

05 Q36

06 REINFORCING BATTALION

07 MANUEVER UNIT

INFO (DIGIT 5,6)

01 MISSION TO FSCORD

02 ADJUSTMENT OF ROUNDS

03 TGT NOT WORTH ENGAGING- CANCEL MISSION

04 MISSION RECEIVED

05 ADJUSTMENT RECEIVED

06 SPLASH

07 BATTERY IS MOVING

08 END OF MISSION

09 INCOMING

10 INCOMING RECEIVED

11 FINAL PROTECTIVE FIRE MISSION

12 TRIGGER MISSION

13 COPPERHEAD MISSION

14 ROUNDS COMPLETE

15 DUPLICATE MISSION IS CANCELLED

16 MISSION IS PREEMPTED

17 SOURCE OF INCOMING

18 FIRE FOR EFFECT

19 BATTERY OR GUN IS READY TO FIRE

20 SHOOT THE MISSION

21 "DESIGNATE" (LASE THE TARGET)

22 BATTERY HAS CLOSED ON POSITION

DOING.CLUSTERS

This routine models the actions of an FO in the development of a multiple target of opportunity. The routine first determines if there are any targets in the FO's target list. If so it then screens any target elements which lie within an active trigger area. The remaining targets elements are then aggregated into groups or clusters which are further assigned a priority based on the number/type/weight of elements in the cluster and its distance from the FO. The cluster with the highest priority is selected as the target of a 'routine' fire mission. Depending on its location relative to the others, a target may be included in more than one cluster. The actual 'clustering' algorithm involves setting a box of dimension BOX.TOLERANCE (user input) around a single target element, with the box centered on that element. The next element of the FO's target list is examined. If it lies within the box, the center of the box is moved to a position which bisects a line between the two target elements. Additional elements which lie within the box will similarly modify its position. The box or cluster is given the average speed and direction of its member elements. Target elements which fall outside the original box initiate new clusters.

ELL.CHECK

This routine provides the calculation which determines if a target lies within an elliptically shaped area. A returned value less than or equal to 1.0 indicates the target is within the ellipse.

END.OF.MISSION

This event completes action on a specific mission by removing it from the appropriate FO's set of active missions (FAMLST). It also decrements the appropriate mission queue (MNVR, CBTRY, SEAD) held by the BN FSO.

FA.ASSESS

This routine identifies all target elements, both blue and red, which lie within the lethal area of a volley of artillery fire. Lethality data is then used to assess the damage results for each element. Consideration is given to protection afforded a potential victim due to its defilade position and the cover provided by surrounding micro terrain (rocks, trees, etc.). 'FALED' (user input) is a 6-d lethality array. ATRIT and TALLEY.HIT.STATE are ground model routines which monte carlo to assign type damage (k/f/m/-kill) to the target, aggregate with all previous damage and remove an element from the battle if it is killed outright or if its aggregate damage is above some 'kill' threshold.

FA.CONV

This routine provides the quadratic coefficients which describe the boundary of an elliptical field. In 'STAR', a field ellipse is described by the following parameters: the battlefield coordinates of the center of the ellipse, the lengths of the semi-major and semi-minor axes of the ellipse, and the orientation angle (in degrees measured counterclockwise from East to the major axis). The quadratic equation for the boundary of the ellipse is:
$$FXX(X-XELL.CEN)**2 + FYY(Y-YELL.CEN)**2 + FXY*(X-XELL.CEN)*(Y-YELL.CEN) = 1.0.$$

FADS

This routine computes the distance between 2 elements when given the x and y coordinates of both elements.

FAHIDE

This routine causes dismounted infantry to take cover when artillery rounds impact in their vicinity. It calls event 'HIDE' from the STAR ground model.

FAMSN.ATLIST

This routine prints out selected attributes of a designated 'MISSION' (temporary entity). It also specifies the current target elements contained in the mission.

FA.PURGE

This routine purges a mission list of target elements (stored in array FA.TGT.LIST) which are dead or to which LOS does not exist. It is called only for target of opportunity missions ('doing.clusters') since only they keep a list of targets. To check LOS, routine 'SIGHT' from the ground model is used.

FA.TGT.ERROR

This routine computes four errors in target location. The size and distribution of the errors is a function of the device the FO is using (laser range finder or binoculars). The user specifies the distribution error (deterministic, uniform, or normal) and the parameters of the distribution for initial location (I=1) and subsequent locations (I=2).

FATIME

This routine generates the time required to accomplish different tasks associated with the FA system. It uses a 2-d array called FA.TIME.DELTAS (user input). Each row of the array corresponds to a specific activity such as PDC mission processing or FO calculations. The first element of each row describes the type distribution and may have one of the following values: 1-deterministic, 2-uniform distribution, 3-normal. If the distribution is deterministic, its value is the 2nd element. If the distribution is uniform, elements 2 and 3 are the start and end points. If the distribution is normal the mean is in element 2 and standard deviation is in element 3 of that particular row.

PDC.PROCESSING

This event determines the time (based on mission type) required to compute the mission and then passes the mission to the guns.

FO.ADJUST

This event models the FO's behavior during the adjust fire phase of a mission. One of 4 things may occur. The FO: 1. loses LOS to all targets elements in his cluster (or elements have been destroyed by other weapons). 2. determines the FFE criteria have been met and notifies Pscord 3. determines further adjustment is needed and notifies Pscord 4. has used up allotted adjusting rounds and must end mission.

PSCORD

This event checks each incoming request for fire to see if it is 'priority'. If so, the mission is transmitted directly to the appropriate firing battery(ies) and the mission is logged in the BN PSO's active (manuever) mission queue. For all missions sent to 'PSCORD' a check is made to insure that no two missions have been generated to attack the same target. Ground

model routine 'HOWPAR' is used to determine if the coordinates of the latest target arrays are within a minimum offset distance of any other target array in the respective queue. If so the latest mission is cancelled and the appropriate FO is notified. Once all duplicate missions are screened routine 'MOA' (method of attack) is called. If moa determines the target is not worth engaging, the mission is cancelled and the FO so notified. If MOA determines the mission is valid but there are no firing units available, the mission is put in the BN FSO's queue (MNVR, CBTZY, or SEAD queues as appropriate). If MOA decides the mission is valid and firing units are available the DS.QUEUE, RN.QUEUE, and/or GS.QUEUE are searched as appropriate and specific batteries (if capable) are assigned to the mission.

INCOMING

This event represents the impact of a single volley of artillery fire either in the adjust or the fire for effect phase of a mission. In either case routine PA.ASSESS is called and returns the number of catastrophic and mobility kills. If the mission is in the adjust phase event FO.ADJUST is scheduled.

MOA

This routine determines the (user input) best method of attack for this mission and searches the idle batteries in the various queues to see if the mission can feasibly be engaged in that manner. If so, it returns an 'ANSWER' of 1. If the available units can't achieve the desired MOA, the routine checks whether the user has specified an alternate method of attack, and if that method is feasible. Thus an ANSWER of 2 indicates that the target can be engaged by an alternate method such as using larger caliber weapons in place of smaller caliber (ie, using 8" guns to shoot for 155mm). An ANSWER of 3 indicates that the user has stated that this particular size and category of target is not worth the expenditure of PA resources. An ANSWER of 0 indicates that the mission is not engagable with currently available fire units.

NEW.COORDINATE.SYSTEM

This routine transforms a location in one coordinate system to the corresponding location in a second coordinate system.

NEW.MISSION

This routine creates a 'routine' (as opposed to a 'priority') fire mission using parameter values determined during the last 'DOING.CLUSTERS'. N.PRI is the number of the cluster having the highest 'tactical importance' (importance being a function of the number of

any vehicles, weighting of these vehicles, and cluster nearness to the FO). PRI is the total weight of that cluster identified above.

POSITION.UPDATE

This routine computes the current location of the centroid of a group of target elements associated with a specific routine mission. It calls 'FA.PURGE' which deletes all dead targets or those to which the FO does not have LOS. If all target elements are purged it returns an 'ANSWER' of no. If there is at least one target left the 'ANSWER' is yes. It also computes the speed and direction of the target as perceived by the observer and his prediction of where the target will be in DELTIME seconds from the current simulation time.

PRI.MSSN

This routine creates a 'priority' fire mission when called by either routine 'TRGR' (which indicates targets are in a trigger area) or routine 'URGENT' (which indicates targets are in the FO's FPF ellipse). It then assigns appropriate values to 'MISSION' attributes.

SHEAF

This routine provides a look-up table returning a sheaf boundary as a function of caliber, number of tubes, and type of sheaf.

SKED.ARTY.FIRES

This event impacts scheduled artillery fires (provided by user) in the battle area and calls 'FA.ASSESS' to determine results.

TRGR

This routine checks each element on the FO's target list to determine if it lies within any assigned trigger area. It aggregates by system-type the weighted value and number of all such elements within each trigger area. It also computes the total number of elements and total weighted value for each trigger area. Given the above data the routine selects the trigger area with the heaviest weighted value, extracts the required parameters from that trigger area, and calls routine 'PRI.MSSN'.

UPDATE.CLUSTERS

This event schedules an 'UPDATE.CLUSTERS' in user input seconds if the FO is still alive in the battle. If there are target elements on the FO's target list the routine will check in sequence the FO's FPF ellipse, his trigger areas, and all identified target clusters to determine if a fire mission is warranted. If at any stage a fire mission is indicated the FO's mission list will be incremented and there will be an attempt to transmit a call for fire.

URGENT

This routine checks all elements on the FO's target list to determine if any are in that FO's PPF ellipse. It totals the number of such elements and aggregates their weights. If the total weight in the PPF ellipse is above a threshold number, routine 'PRI.MSSN' is called.

APPENDIX B
CURRENT STAR FIELD ARTILLERY INPUT VARIABLES

AMT.AMMO.TYPES

This variable identifies the number of types (HE, DPICM, etc.) of ammunition used.

AMT.BLUE.BATTERYS

This variable is the number of blue batteries to be simulated.

AMT.CALIBERS

This variable is the number of different calibers of ammunition used.

AMT.FA.TIME.DELTAS

This variable is the number of different time parameters listed in array FA.TIME.DELTAS. (See routine FATIME).

AMT.RED.BATTERYS

This variable is the number of red batteries to be simulated.

BOX.TOLERANCE

This variable is the length of one side of the box used by the FO in the routine 'DOING.CLUSTERS'.

CALIBER (I)

This array contains the permanent attribute of BATTERY which identifies the caliber of a battery's guns. It may be assigned one of the following values: 1-155mm 2-203mm 3-GSRS 4-152mm 5-122mm MRL.

COLOR1 (I)

This array contains the permanent attribute of BATTERY which identifies a specific battery as blue or red.

COLOR2 (I)

A permanent attribute of each fire direction center. It identifies the FDC as blue or red. This array contains the permanent attribute of FIRE DIRECTION CENTER which identifies a specific FDC as blue or red.

DIRCTR (I)

This array contains the permanent attribute of FORWARD OBSERVER which identifies the fire direction center that provides direct support for a specific FO.

FA.BN (I)

This array contains the permanent attribute of FIRE DIRECTION CENTER that identifies to which FA Bn a specific battery FDC belongs.

FALED (I-N)

This is a six dimensional lethality array used in artillery impact assessments.

FA.MINV

This variable is the minimum number of target

elements (vehicles) which must lie within a cluster before a fire mission is generated.

FAMSN (I)

This array contains the permanent attribute of BATTERY which indicates if a unit is in a direct support, general support, or reinforcing role.

PA.TIME.DELTAS (I,J)

This is a 2-dimensional array containing the parameters that characterize the time distributions for all artillery tasks. (See routine FATIME).

FATM

This is the time variable used in event 'SKED.ARTY.FIRES' which identifies the time until the event is rescheduled.

PAWGT(I,J,K)

This 3-dimensional array holds the weighting (value) factors placed on target elements of different types. The priority/ranking of an individual 'cluster' is a function of the weighting factor of the target elements, the number of target elements (vehicles), and the distance of cluster center from the FO.

FO.VEHICLE

This variable is the identification number of the 'TANK' or 'UNIT' to which the FO is assigned. It permits use of all 'TANK' attributes.

FPPN

This variable is the number of target elements (vehicles) needed in an FO's FPP ellipse in order to activate a priority mission.

FWD.OBS.MSN.TOLERANCE

This variable identifies the minimum distance two active fire missions can be apart (as measured from the center of their cluster) before being classified as the same mission.

FXHID

This time variable is used in routine 'FAHIDE' and identifies the length of time after the last volley of artillery fire that a dismounted infantry element will remain covered.

MAXMAJ

This variable is the length of the largest semi-major axis of any FO's FPP or trigger ellipse.

MAX.RANGE(I)

This 1-dimensional array holds the maximum range (in meters) of artillery weapons assigned to a specific battery.

MISS.TOLERANCE

This variable identifies the minimum adjusting distance allowed in an adjust-fire phase. Values less than this will cause the mission to go to the fire-for-effect phase.

MOACBTRY (I,J,K,L)

This is a 4-dimensional array which defines the method of attack for counterbattery missions. The dimensions are defined as follows: I- vector which represents the predominant vehicle type in the target. J- vector which represents the physical size (area) of the target. K- vector which represents the total number of vehicles in the target. Elements L(1)-L(4) represent the following 'best method of attack': L(1)- the number of firing units to shoot the target upon FFE. L(2) the number of volleys to shoot upon FFE. L(3) the caliber of the guns. L(4)- the type ammunition. Elements L(5)-L(8), L(9)-L(12), ... L(n)-L(n+4) may be included by the user to specify alternative (next best) methods of engaging a specific type target. (see routine MOA)

MOAMVR (I,J,K,L) Method of attack for maneuver missions. See MOACBTRY above.

MOASEAD (I,J,K,L) Method of attack for SEAD missions. See MOACBTRY above.

MSN.PRTY

This is a permanent attribute of the FO. A value of 1 implies that the FO has a maneuver mission, 2 indicates a CBTRY mission, and 3 indicates a SEAD mission.

N.ADJ.RDS

This variable is the maximum number of adjusting rounds available to the FO. If the FO requires this number without going to fire-for-effect his mission will be cancelled.

N.BATTERY

This variable is the total number of batteries to be simulated, both red and blue.

NCALS

This variable is the number of different caliber weapons to be simulated for both blue and red.

N.PDC

This variable is the total number of fire direction centers (permanent entity) to be simulated.

N.FO

This variable is the total number of forward observers (permanent entity) to be simulated for both blue and red.

NMOA

This variable is the total number of methods of attack to be used.

NO.RANGE.BANDS

This variable is the number of range bands in array 'RANGE.BANDS'.

N.TO.FIRE

This variable is the number of volleys to be fired in an FPF.

NTRIG This variable is the number of trigger areas to be simulated.

NUM.GUNS (I) This array holds a permanent attribute of BATTERY which identifies the number of guns in a specific battery.

RANGE.BANDS (I,J) This is a 2-dimensional array containing the breakpoints of the piecewise linear approximation to the impact point dispersion curves.

RATE.OF.FIRE This is a permanent attribute of battery. It represents the maximum rate of fire of the battery in rounds per minute.

SIGMA.DPICM (I,J,K) This is a 3-dimensional array containing the parameters of the normal distribution that characterize the round dispersion about the impact point.

SJFPF This variable is the length of the semi-major axis of an FPF ellipse.

SNFPF This variable is the length of the semi-minor axis of an FPF ellipse.

TGT.ACQ.ERROR This is a 2-dimensional array containing the parameters that characterize the error distributions of the FO's target acquisition devices-different for each device.

THFPF This variable represents the angle (in radians) of the semi-major axis of the FPF measured with a positive value in the counter clockwise direction from East.

TRAVEL.TIME.ARRAY (I,J) This is a 2-dimensional array containing the average velocity at two-thirds the maximum range of the system for each ammunition/ weapon combination.

TRIG.A (I) This is a 1-dimensional array which contains the number of trigger areas to be simulated and thirteen additional information elements for each specific trigger area. The first 14 elements are defined as follows: 1-the number of trigger areas to be simulated, 2- the x coordinate of the centroid of the first trigger area, 3- the y coordinate of the centroid of the first trigger area, 4- the length of the major axis of the trigger ellipse, 5-the length of the minor axis of the trigger ellipse, 6-the orientation angle of the trigger ellipse, 7- the minimum number of vehicles required to lie within a trigger area in order to activate it, 8- the minimum weight (summation of the

weighting factors for each enemy vehicle within the trigger area) 9- the type of ammunition to be fired for this trigger area, 10- the number of firing units to engage this trigger area, 11- the pointer to TRIG.B (indicates which specific fire units will engage this trigger area), 12- the x coordinate of the aim point of the volley to be fired, 13- the y coordinate of the aim point of the volley to be fired, 14- the number of volleys for each firing unit to fire. TRIG.A(15)-TRIG.A(27) and TRIG.A(28)-TRIG.A(40) are for additional trigger areas and would contain the same type information as held in TRIG.A(2)-TRIG.A(14).

TRIG.B (I)

This is a 1-dimensional array which holds the firing units designated to shoot when a specific trigger area is activated.

TYPE

This variable identifies the type of device used by the FO to locate targets. It is usually a laser range finder (1) or binoculars (2).

X.CUR2 (I)

This 1-dimensional array holds in each cell the value of this permanent attribute of the FIRE DIRECTION CENTER. It is the x coordinate of the FDC location.

X.CUR1 (I)

This 1-dimensional array contains values for the permanent attribute BATTERY which represent a battery's current x coordinate location in the battlefield coordinate system.

X.PPF

This variable is a permanent attribute of an FO. It identifies the x coordinate of the centroid of that FO's final protective fire ellipse.

X.TO.FIRE

This variable is a permanent attribute of an FO. It is the x coordinate of the actual center of impact of an PPF volley.

NOTE: For y coordinate variables see corresponding x coordinate variables above.

APPENDIX C

KEY COPPERHEAD ENTITIES

ROUND

(ROUND is a temporary entity which belongs to CHSET)

ACQ.TIME

This attribute is the simulation time in which round acquires the target.

FIRE.TIME

This attribute is the simulation time in which the round is fired.

GAP

This attribute is the I value in the array R.HT.FP(I,J,K). It relates to gun-aim point range for the round.

HEIGHT

This attribute is the J value in array R.HT.FP(I,J,K). It relates to the altitude of the round in which acquisition is made.

IMPACT.TIME

This attribute is the simulation time in which the round is scheduled to impact.

LASE.ON

This attribute is a 0/1 value to indicate if the Forward Observer's laser is off/on for this round.

RD.NAME

This attribute is an integer value which acts as the round's name (sequence number) for a specific mission.

IGT.ID

This attribute holds the name of the specific target vehicle being designated by the FO.

X.DEL

This attribute is the distance measured perpendicularly from the gun-aim point line that an unguided round would fall from the aim point of the mission. It is deflection error.

Y.DEL

This attribute is the distance measured along the gun-aim point line that an unguided round would fall from the aim point of the mission. It is the range error.

MISSION

(MISSION is a temporary entity currently contained in the Field Artillery module. Of its approximately 56 attributes, those below are specifically related to Copperhead)

AMMUNITION.TYPE

This attribute is given an integer value of 6 if the mission is Copperhead.

DIRECTION

This attribute is used in a Copperhead mission as a counter. Its name is changed in a Preamble "define to mean" statement to NUM.FIRED(M) in order to provide greater clarity to the program analyst. It provides a value for RD.NAME in routine CH.FIRE.

GT.INITIAL.RG

This attribute holds the gun-aim point range.

NUM.FIRED

See attribute DIRECTION above.

MSN.TYPE

This attribute is given an integer value of 1 if the mission is Copperhead.

PHASE

This attribute, when given a value of 'FFE' in event CH.FIRE, will later cause end of mission processes to begin in the calling event, BTRY.FIRING.

THETA

This attribute holds the angle made by the gun-aim point line and the East-West line. It is used to convert from a gun-aim point coordinate system to a battlefield coordinate system.

VOLLEYS.TO.FIRE

This attribute contains the number of rounds selected by the FO to fire for this mission. It is directly related to the number of targets the FO predicts will be in the footprint in the system response time.

X.CUHD

This attribute identifies the x coordinate of the new center of a maneuver footprint. It is given a value when the FO attempts to switch targets after beginning a lase.

X.FUTURE.LOC

This attribute is the x coordinate of the aim point for the mission.

Y.CUHD

This attribute identifies the y coordinate of the new center of a maneuver footprint. It is given a value when the FO attempts to switch targets after beginning a lase.

Y. FUTURE. LOC

This attribute is the y coordinate of the aim point for this mission. In this case, the aim point refers to the coordinates held in array AIM.PT. These coordinates are the desired impact point called by the FO and not the ballistic unguided point of impact for a specific round.

APPENDIX D

COPPERHEAD ARRAYS AND VARIABLES

AIM.POINT

This two dimensional array contains coordinates of all preplanned Copperhead targets available for use during the simulation. (See figure 10.) These targets are established by the user and each represents the aim point of the center of a Copperhead maneuver footprint. The aim points are arranged in sets, each set being associated with a position area which may possibly be occupied by the forward observer at some time in the battle. (The term forward observer refers to whatever member of the First team is actually operating the GVILD). Because of the dynamic nature of STAR, the user will have significant flexibility in selecting possible position areas. One approach would be to select Company Headquarters' position areas (platoon size) at each coordination line throughout the company's area of operation in the Main Battle Area. A vehicle location with the best line of sight could be specified for the forward observer within each such position area. After making these selections the user would then simply identify Copperhead aim point coordinates to be activated should the FO arrive at a specific position area. In this sense the user performs the function of a Bn Fire Support Officer who is preparing a fire support plan.

Each aim point will be identified by the user as priority or normal Copperhead target. The user must insure that priority assignments along any particular coordination line do not overwhelm the capability of the supporting artillery unit. The position area actually occupied by the FO at any time during the simulation is held in attribute AREA.START - which is an attribute of the TANK or UNIT associated with the FO.

CH.A

This global variable is a user input which identifies the time interval between firings of a Copperhead round during a multiple round mission. It is normally 20 seconds.

CH.B

This global variable is a user input which identifies the time between impact of a Copperhead round and the time the FO must begin lasing for a subsequent round during a multiple round mission. It is normally about 7 seconds.

CH.C

This global variable is a user input which identifies the difference between the time of impact of the initial round of a Copperhead mission and a time earlier when the FO is alerted by a 'designate' message from his FDC.

CH.PKH

This two dimensional array provides conditional kill probabilities as a function of weapons type and defilade status. (See figure 11.) The array is used by routine CH.HIT in conjunction with two attributes of the specific Copperhead target (called a TANK or UNIR in the computer code). One attribute is 'WPN.TYPE(TANK)', which identifies the target as a particular type of weapon/vehicle, such as a P-72 tank, BMP, ZSU-23, etc. The other attribute is DEFNUM(TANK) which defines the defilade status of the target at the current simulation time. A DEFNUM(TANK) value could indicate that target was fully exposed, in turret defilade, or in full defilade. (An enemy vehicle in full defilade position would theoretically not be selected by the program logic as a target for Copperhead). The conditional probabilities are provided by the Army Materiel Systems Analysis Activity and include catastrophic kill, mobility kill, and firepower kill.

CH.NORM.MAX

This global variable is a user input which indicates the upper bound of a normal response 'time band'.

CH.NORM.MIN

This global variable is a user input which indicates the ideal expected system response time of a normal Copperhead mission. It is the lower bound of a normal response 'time band' used in routine COPPERHEAD. System response time is defined as the time between the initiation of the forward observer's call for fire and the impact of the first round. It is used by the FO, along with apparent speed and direction of a target, in computations to predict the future location of that target. System response time is not a precise value but only a planning figure developed by the FO based on his experience with his supporting artillery unit.

CH.MINV

This global variable is a user input which establishes a minimum threshold value necessary to generate a Copperhead mission. The forward observer predicts the number of enemy vehicles which will be within a specific footprint for a given response 'time band'. He then compares this number of vehicles (or a corresponding target weight) against the threshold value required to start a mission. If this number is greater than the threshold number, the logic segment will take action to generate a mission.

CH.PRI.MAX

This global variable is a user input which indicates the upper bound of a priority response 'time band'.

CH.PRI.MIN

This global variable is a user input which indicates the ideal expected system response time of a priority Copperhead mission. It is the lower bound of a priority response 'time band' used in routine COPPERHEAD. System response

time is defined as the time between the initiation of the forward observer's call for fire and the impact of the first round. It is used by the FO, along with apparent speed and direction of a target, in computations to predict the future location of that target. System response time is not a precise value but only a planning figure developed by the FO based on his experience with his supporting artillery unit.

DEL.T

This global variable is a user input which is used in event CH.LASE to regularly reschedule the event.

Et

This global variable is a user input which measures the threshold energy density at the seeker aperture. It is a technical characteristic of the round.

GAM

This variable is the atmospheric attenuation coefficient which is a function of visibility (12, p 12).

NUMBER.ROUNDS

This one dimensional array contains the number of rounds to be fired for a specific Copperhead mission. It will be a user input array and will be based on the forward observer's prediction of the potential targets to be within a specific Copperhead footprint at the time of system response.

P

This global variable is a user input which measures target reflectivity. It is used in the acquisition calculations conducted in routine CH.ACQUIRE.

R.HT.FP

This three dimensional array provides the key data for the impact phase of the model. (See figure 12.) Its inputs will be data very similar to that used by the Army Materiel Systems Analysis Activity in the PAM model. The first dimension of the array represents the set of gun-aim point ranges of a Copperhead mission, from the minimum range to the maximum range, at even one kilometer intervals. The next dimension represents time before impact of the round, displayed in one second increments from T-13 to T-1 seconds. The final dimension contains the appropriate round altitude for the respective gun-aim point range given a specific angle of fall and time before impact. Also included are the specific parameters of the footprint associated with the above variables.

TEMPLATE

This two dimensional array contains the same basic data that in the real world is provided by

the forward observer's Copperhead template. (See figure 13.) This is a plastic device displaying individual footprint ellipses of various sizes with associated cloud heights and gun-aim point ranges. The template is used by the FO to draw footprint overlays on his tactical map. It enables him to envision the boundaries of a footprint with respect to the actual terrain near his present position. In a general sense the FO's template allows him to determine whether or not there will be in system response time a sufficient number of enemy targets within the maneuver capability of a Copperhead round fired at a pre-planned aim point.

TDF

This two dimensional array contains the time of flight of a Copperhead projectile given a specific gun-aim point range and an angle of fall. Data for the array is available at the Army Materiel Systems Analysis Activity.

113	1	X11	Y11	0	X12	Y12	0	X13	Y13
136	0	X21	Y21	0	X22	Y22	0	X23	Y23
186	0	X31	Y31	1	X32	Y32	-	-	-
208	0	X41	Y41	0	X42	Y42	0	X43	Y43
210	1	X51	Y51	1	X52	Y52	-	-	-

...

FP#2 ellipse Y-coord

FP#2 ellipse X-coord

Priority/Normal (1/0) FP#2

FP#1 ellipse Y-coord

FP#1 ellipse X-coord

Priority/Normal (1/0) FP#1

FO position area

FIGURE 10. ARRAY AIM.PT DIAGRAM

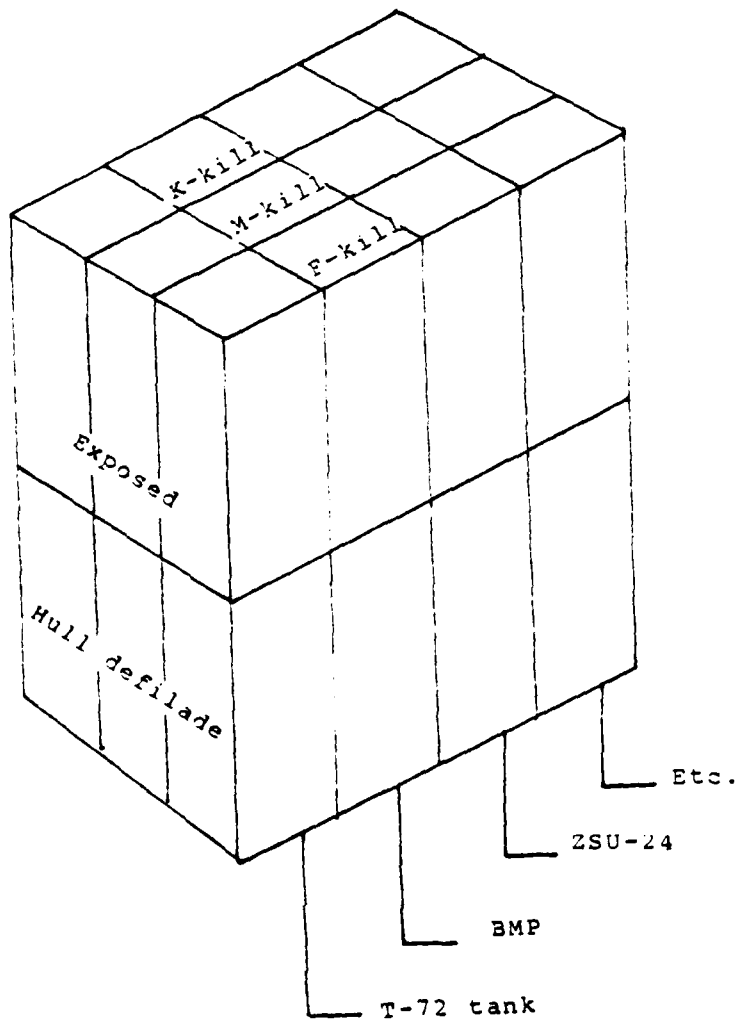


FIGURE 11. ARRAY CH.PKH DIAGRAM

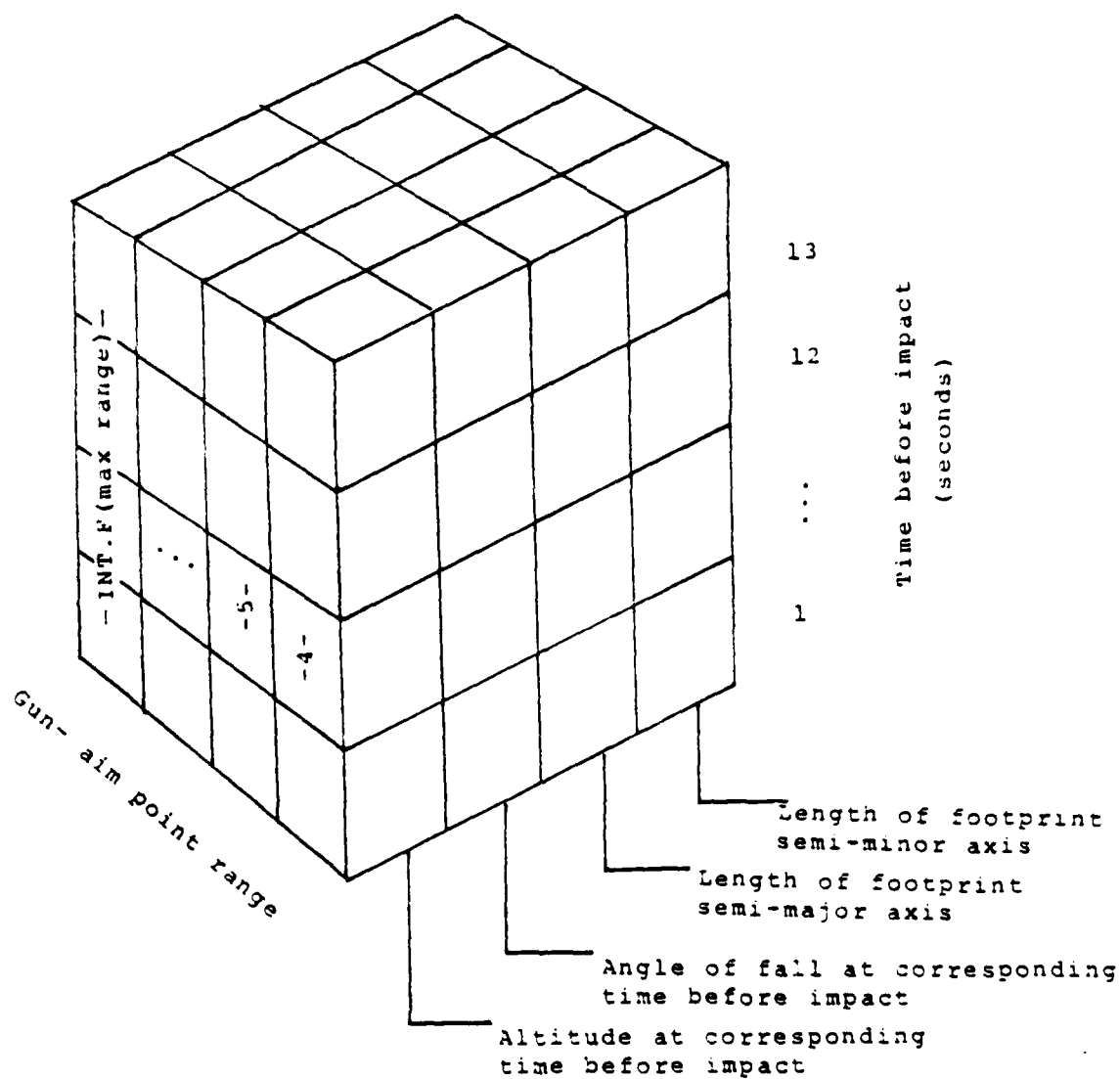


FIGURE 12. ARRAY R.HT.FP DIAGRAM

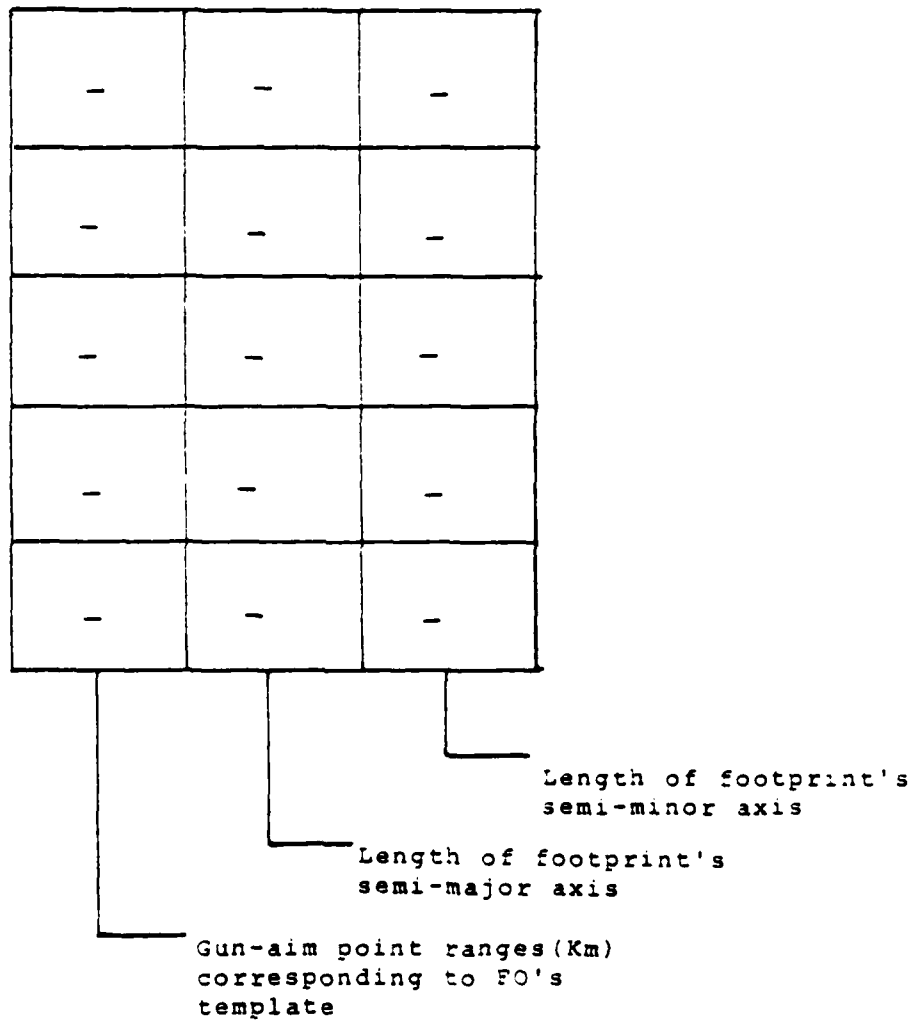


FIGURE 13. ARRAY TEMPLATE DIAGRAM

LIST OF REFERENCES

1. Carpenter, Howard J. and Thurman, Edward E. Parametric Simulation of Infantry Tactics and Equipment (Dismounted STAR), M.S. Thesis, Naval Postgraduate School, Monterey, June 1980.
2. Haislip, William A. Jr. Communication/Electronics Warfare Modules for Simulation of Tactical Alternative Responses (STAR), M.S. Thesis, Naval Postgraduate School, Monterey, March 1979.
3. Department of the Army, FM 6- 30 (Copperhead), Observer Procedures for G/VLLD and Copperhead Missions, Final draft.
4. Doughty, M.A., Holder, L.D., "Images of the Future Battlefield", Military Review, p.58, Jan 1978.
5. Department of the Army, TC 6- 30-1, The Copperhead GLLD System, (Undated Draft).
6. Hartman, J.K. Ground Movement Modelling in the STAR Combat Model, Naval Postgraduate School Technical Report NPS 55-80-021, Monterey, California, May 1980.
7. Hartman, J.K. Parametric Terrain and Line of Sight Modelling in the STAR Combat Model, Naval Postgraduate School Technical Report NPS-55-79-018, August 1979.
8. Hartman, J.K. The STAR Field Module, Naval Postgraduate School Technical Report NPS 55-80-023, June 1980 (p. 7-17).
9. Kelleher, Edward P. Jr. Simulation of the Tactical Employment of Field Artillery, M.S. Thesis, Naval Postgraduate School, Monterey, California, December 1977.
10. Lewis, C.L., A.G. Nichols, and A.W. Lee, Users Guide for the Phase I Laser Designator/Weapon System Simulation (LDWS) of the Copperhead Guided Projectile System, Vol.1, Technical Report RG-77-25. US Army Missile Research and Development Command, Redstone Arsenal, Al., July 77,.
11. Parry, S.H. and Kelleher, E.P. The Tactical Parameters and Input Requirements for the Ground Components of the STAR Combat Model, Naval Postgraduate School Technical Report NPS 55-80-021 Monterey, May 1980.

12. Starks, Michael A. Monte Carlo Model for Determining Copperhead Probability of Acquisition and Maneuver. US Army Materiel Systems Analysis Activity, DA Project No. 1R765706M541.
13. Starner, Steven G. A Two Sided Field Artillery Stochastic Simulation. M.S. Thesis, Naval Postgraduate School, Monterey, California, March 1979.
14. Stockfish, J.A. Models, Data, and War: A Critique of the Study of Conventional Forces. US Air Force Project Rand Report R-1525-PR, March 1975.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4. Professor James K. Hartman Code 55Hh Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
5. Professor S. H. Parry, Code 55Py Department of Operations Research Naval Postgraduate School Monterey, California 93940	5
6. Professor Arthur L. Schoenstadt, Code 53Zh Department of Mathematics Naval Postgraduate School Monterey, California 93940	5
7. Office of the Commanding General U.S. Army TRADOC ATTN: General Donn A. Starry Ft. Monroe, Virginia 23651	1
8. Headquarters U.S. Army TRADOC ATTN: ATCG-T (BG Don Morelli) Ft. Monroe, Virginia 23651	1
9. Headquarters U.S. Army TRADOC ATTN: Seymour Goldberg Combat Developments Ft. Monroe, Virginia 23651	1
10. Headquarters U.S. Army TRADOC Attn: Director, Manuever Directorate Combat Developments (COL Schurtz) Ft. Monroe, Virginia 23651	1
11. Mr. Walt Hollis Deputy Under Secretary of the Army (Operations Research) Department of the Army, The Pentagon Washington, D.C. 20310	1

12. LTG William Richardson 1
 Commanding General
 U.S. Army Combined Arms Center
 Ft. Leavenworth, Kansas 66027
13. Director 1
 Combined Arms Combat Development Activity
 Attn: Mr. Lee Pleger
 Ft. Leavenworth, Kansas 66027
14. Director 1
 Combat Analysis Office
 Attn: Mr. Kent Pickett
 U.S. Army Combined Arms Center
 Ft. Leavenworth, Kansas 66027
15. Command and General Staff College 1
 Attn: Education Advisor
 Room 123, Bell Hall
 Ft. Leavenworth, Kansas 66027
16. Dr. Wilbur Payne 1
 Director
 U.S. Army TRADOC Systems Analysis Activity
 White Sands Missile Range, New Mexico 88002
17. Headquarters 1
 Department of the Army
 Office of the Deputy Chief of Staff
 for Operations and Plans
 Attn: LTG Glenn Otis
 Washington, D.C. 20310
18. Commander 1
 U.S. Army Concepts Analysis Agency
 8120 Woodmont Avenue
 Attn: MOCA-SMS (Cpt Steve Shupack)
 Bethesda, Maryland 20014
19. Director 1
 U.S. Army Night Vision & Electro-optical Lab.
 Attn: DEL-NV-VI (Mr. Bob Hermes)
 Ft. Belvoir, Virginia 22060
20. Director 1
 U.S. Army Materiel Systems Analysis Activity
 Attn: Mr. Will Brooks
 Aberdeen Proving Grounds, Maryland 21005
21. Director 1
 U.S. Army Materiel Systems Analysis Activity
 Attn: Mr. Michael Starks
 Aberdeen Proving Grounds, Maryland 21005
22. Director 1
 Armored Combat Vehicle Technology Program
 Attn: COL Len Fleming
 U.S. Army Armor Center
 Ft. Knox, Kentucky 40121
23. Director 1
 Combat Developments, Studies Division
 Attn: Maj W. Scott Wallace
 U.S. Army Armor Agency
 Ft. Knox, Kentucky 40121

24. Commandant 1
U.S. Army Field Artillery School
Attn: ATSP-MBT (Cpt Steve Starner)
Ft. Sill, Oklahoma 73503
25. Director 1
Combat Developments
U.S. Army Aviation Agency
Ft. Rucker, Alabama 36360
26. Director 1
Combat Developments
U.S. Army Infantry Agency
Ft. Benning, Georgia 31905
27. Director 1
Missile Intelligence Agency
Attn: ADA Tactics (Cpt. E.G. Hagewood)
Redstone Arsenal, Alabama 35809
28. Director 1
Combat Developments
Attn: Cpt William D. Meiers
U.S. Army Air Defense Agency
Ft. Bliss, Texas 79905
29. Commander 1
U.S. Army Logistics Center
Attn: ATCL-OS (Mr. Cammeron/Cpt Schuessler)
Ft. Lee, Virginia 23801
30. Commander 1
USANMCS
Attn: ATSK-CD-CS (Mr. Lee/Mr. Marmon)
Redstone Arsenal, Alabama 35809
31. Director 1
U.S. Army Materiel Systems Analysis Activity
Attn: DRXSY-AA (Mr. Tom Coyle)
Aberdeen Proving Grounds, Maryland 21005
32. Director 1
Army Models Management Office
Attn: LTC(P) David Robinson
USACAC
Ft. Leavenworth, Kansas 66027
33. Chief 5
TRADOC Research Element Monterey (TREM)
Naval Postgraduate School
Monterey, California 93940
34. Office of the Deputy Chief of Staff 1
for Combat Developments
U.S. Army TRADOC
Attn: MG Carl Vuono
Ft. Monroe, Virginia 23651
35. Deputy Commanding General 1
Combined Arms Combat Development Activity
Attn: ATZL-CA-DC
Ft. Leavenworth, Kansas 66027

36. Director 1
Combat Developments
Attn: COL Pokorney
U.S. Army Field Artillery Center
Ft. Sill, Oklahoma 73503
37. Director 1
USATRASANA
Attn: Mr. Ray Heath
White Sands Missile Range, New Mexico 88002
38. Commander 1
U.S. Army ADMINCEN
Attn: AT2I-CD-AD (Maj Cochard)
Ft. Benjamin Harrison, Indiana 46216
39. Acting Director 1
CASAA
Attn: ATZL-CAR-MD (Cpt Reischl)
Ft. Leavenworth, Kansas 66027
40. Cpt Thomas E. Cahill 1
C/O Louis Graziano
PO Box 143
Highland Falls, New York 10928

DATE
FILMED
-8